Linux Newbie Administrator Guide

Table of Contents

Linux Newbie Administrator Guide 1
Introduction 1
Linux Newbie Guide: Linux Benefits 3
0.1 Fundamentally, why Linux? 4
0.2 Is Linux for me? 4
0.2a Linux is difficult for newbies. 4
0.3 What are the Linux benefits? 4
0.4 I don't believe in free software, etc. 6
0.5 "There ain't no such thing as a free lunch" 6
0.6 I need high security. With commercial software, I can sue them if things go wrong. . 7
0.7 I need standards. Big software corporations (Microsoft) provide standards. . . 7
Linux Newbie Guide: Before Installation 9
1.1 Which Linux distribution should I use? 10
1.2 What are the Linux hardware requirements? 11
1.3 Will my hardware work under Linux? 13
1.4 How do I download Linux? 13
1.5 How do I get a Linux CD? 14
1.6 I have RedHat CD but no install floppy. What do I do? 14
1.7 What do I need to read before installation? 16
1.8 Can I have MS Windows and Linux installed on the same computer? 16
1.9 How do I partition my hard drive? 17
1.10 The MS Windows partition occupies my whole harddrive. Can I shrink/split it without a re-install? 19
1.11 How do I start the installation? 19
1.12 Is the Linux installation difficult? 19
1.13 I finished the installation. How do I log-in for the very first time? 20
Linux Newbie Guide: Resources, Help And Links 21
2.1 Any Linux reading materials? 21
2.2 Is there a help command? 22
2.3 Newsgroups 23
2.4 Any Linux Internet links? 24
Linux Newbie Guide: Basic Operation FAQ 27
3.1 Basics 28
3.1.1 Filenames 28
3.1.2 What are the different directories for? 30
3.1.3 How do I run a program? 31
3.1.4 How can I change the PATH? 33
3.1.5 How can I shutdown my computer? 34
3.1.6 How do I deal with a hanged program? 35
3.2 Users, passwords, file permissions, and security 38
3.2.1 Home directories, root, adding users 38
3.2.2 About password security 39

3.2.3 I forgot the root password 39
3.2.3a I forgot my user password. 41
3.2.4 I have file permission problems. How do file ownership and permissions work? . 41
3.2.5 My mp3 player chokes. The sound is kind of interrupted (how to set suid). . . 45
3.3 Job scheduling with "at", "batch", and cron 46
3.3.1 How do I execute a command at specified time (using "at" or "batch")? . . 46
3.3.2 How do I set up cron? 47
3.4 Swap space 48
3.4.1 Swap partitions 49
3.4.2 Swap files 49
3.5 Shell 50
3.5.1 What is a shell and do I want to use a different one? 50
3.5.2 How do I write a simple shell script? 51
3.6 Simple programming 52
3.6.1 How do I write a simple Python program? 52
3.6.2 How do I write a simple GUI program (using Tk)? 52
3.6.3 How do I write a simple perl script? 53
3.6.4 How do I compile a simple C program? 54
3.7 How do I install a program I downloaded from the Internet? 54
Linux Newbie Guide: Administrator FAQ 57
4.1 LILO issues 58
4.1.1 Linux cannot detect all my memory 58
4.1.2 LILO displays only LI (or LIL) and hangs 59
4.1.3 How can I change the operating system that LILO boots on default? . . . 61
4.1.4 The LILO prompt stays too short (or too long) on the screen during the bootup . 61
4.1.5 Uninstalling Linux 61
4.2 Accessing my drives 61
4.2.1 Where are my drives? 61
4.2.2 How can I access my CDROM? 62
4.2.3 How to mount a floppy, zip drive, DOS/Windows partition, or a network drive? . 63
4.2.4 How to mount a remote MS Windows filesystem through Samba? 65
4.2.5 Any quick way to access a file on a DOS/Windows floppy? 65
4.2.6 Mounting works when I am root. Can a normal user mount? 66
4.2.7 Mounting command is too long, how can I simplify it with an alias? . . . 68
4.2.8 Can I automount? 68
4.2.9 How do I get my parallel-port (external) Zip drive recognized? 69
4.2.10 Can I set 32-bit hard drive I/O? 71
4.3 Working with X-windows 72
4.3.1 How do I install kde on RedHat 5.2? 72
4.3.2 How do I setup video card, monitor and mouse for the X-server? 74
4.3.3 How to X-window remotely? 74
4.3.4 Can I have multiple sessions of X running at the same time? 76
4.3.5 Can I have a GUI login prompt? 76
4.4 Printer and soundcard 77
4.4.1 How to setup my soundcard? 77
4.4.2 How do I setup my printer? 78

4.4.3 Word Perfect 8 does not have a driver for my printer 80
4.5 Setting up a network 80
4.5.1 Would it be worth it to set up my home network? 80
4.5.2 How to set up my home network? 80
4.5.3 I have problems configuring my ppp dial out 83
4.5.4 How to browse the net from my networked computer without a modem? . . 85
4.5.5 How to use Samba? 86
4.5.6 Sendmail 88
4.5.7 Simple web server (running Apache) 89
4.5.8 Simple ftp server 90
4.5.9 How can one access my computer from the outside world when I am on the net using my phone ppp conection? 91
4.5.10 Can my home computer get hacked? 92
Linux Newbie Guide: Kernel Upgrade 94
5.1 Introduction 95
5.2 What is kernel? 95
5.3 Why upgrade? 95
5.4 Where can I get the new kernel? 95
5.5 Should I download full source or the patch only? 100
5.6 Unpacking the download file 100
5.6.1 Unpacking the patch 100
5.6.2 Unpacking the full source 100
5.7 Compiling new kernel 101
5.7.1 Using the patch 101
5.7.2 Using the full source 102
5.8 Installing the new kernel 102
5.9 What about next kernel upgrade? 104
5.10 Notice to Red Hat users 105
5.11 About the author 105
5.12 Changes 105
Linux Newbie Guide: Shortcuts And Commands 106
6.1 Linux essential shortcuts and sanity commands 107
6.2 Common Linux commands--system info 109
6.3 Basic operations 110
6.4 Process control 112
6.5 Basic administration commands 113
6.6 Networking tools 116
6.7 Linux built-in developer tools and programming languages 117
Linux Newbie Guide: Linux Applications 119
7.1 Word Perfect 8 for Linux 120
7.2 Star Office Suite 120
7.3 Applixware 121
7.4 AbiWord 121
7.5 Spreadsheet 121
7.6 Databases 122
7.7 CAD 122

7.8 Netscape and Lynx (web browsers) 123
7.9 Klyx and Latex 124
7.10 Writing CD-Rs (cdrecord and cdparanoia) 126
Linux Newbie Guide: Acknowledgments 132
Acknowledgments 132
Other matters 132
Our rewards and testimonials 132

THE HOBBY OF THE KLIMAS FAMILY

ALWAYS UNDER DEVELOPMENT

LINUXNEWBIEADMINISTRATORGUIDELNAG

by Stan, Peter, Marie, and Barbara Klimas

Easy Answers to the Questions Frequently Asked by the Beginner Users of Linux who Install Linux on their Home Computers or Administer their Home Network

Distributed under the General Public Licence http://www.gnu.org/copyleft/gpl.html. Your feedback, comments, corrections, and improvements are appreciated. Send them to bklimas@magma.ca

(Generated from the html version of the Linux Newbie Administrator Guide)

Introduction

We (the Klimas family) are relative Linux newbies (with Linux since Summer 1998). We run RedHat mostly -> the solutions might not be directly applicable to other Linux distributions (although most of them probably will). Hope this helps, we try to be as practical as possible. Of course, we provide no warranty whatsoever!

Part 0: **For the Undecided (Linux Benefits)**

If you wonder what are Linux pros and cons, and whether Linux is for you.

Part 1: **Before Linux Installation**

What distribution should I use, how to obtain it, Linux hardware requirements, how to partition your hard drive, about dual boot, and how to login the very first time.

Part 2: **Linux Resources, Help and Some Links**

If you would like to know how to access the Linux documentation, wonder if Linux has a help command, or wanted to visit some Linux-oriented newsgroups or websites.

Part 3: **Basic Operations FAQ**

After you installed Linux, here are answers to some questions that Linux newbie users/administrators may have when trying to perform every-day tasks: how to run a program, shut down your computer, set up the path, add users, work with file permissions, schedule jobs with "at" and cron, set up the swap space ...

Part 4: **Linux Newbie Administrator FAQ**

More easy answers to questions that Linux newbie administrators frequently encounter: LILO issues, mounting drives, using X-windows, setting up network, ppp connection, printer, soundcard, remote access to your computer ...

Part 5: **How to upgrade the kernel (by Alesh Mustar)**

All you need to know to upgrade the Linux kernel.

Part 6: **Linux Shortcuts and Commands**

Maybe this should come first. A practical selection of Linux shortcuts and commands in tabular form. Perhaps this is everything what a newbie Linuxer really needs.

Part 7: **Essential Linux applications (proprietary or not)**

Essential and/or famous Linux applications are mentioned with some hint/comments (this is just started and will be expanded).

Acknowledgments

The master copy of this page: http://sunsite.auc.dk/linux-newbie/ (Denmark, Europe)
Mirror: http://www.magma.ca/~bklimas/ (Ontario, Canada)
Mirror: http://www.linsup.com/newbie/ (Australia)
A Russian translation (ver. 0.10): http://www.college.balabanovo.ru/rider/book/(Russia)
A Chinese translation of the "Linux Shortcuts and Commands" (ver. 0.32) is available here.

Penguin from Deep River, Ontario :-)
We are proud Canadians of Silesian background.

LINUX NEWBIE ADMINISTRATOR GUIDE

ver. 0.57 1999-11-15

Part 0: For the Undecided (Linux Benefits)

by Stan and Peter Klimas

Answers to Some Frequently Asked Linux Questions

Distributed under the General Public License http://www.gnu.org/copyleft/gpl.html. Your feedback, comments, corrections, and improvements are appreciated. Send them to bklimas@magma.ca

---

Quick site navigation:

Start: Linux Newbie Administrator Guide
Part 0: >For the Undecided (Linux Benefits)<
Part 1: Before Linux Installation
Part 2: Linux Resources, Help and Some Links
Part 3: Basic Operations FAQ
Part 4: Linux Newbie Administrator FAQ
Part 5: How to Upgrade the Kernel
Part 6: Linux Shortcuts and Commands
Part 7: Essential Linux applications (proprietary or not)

Contents of this page:

0. For the Undecided (Linux Benefits)

0.1 Fundamentally, why Linux? [p 4]
0.2 Is Linux for me? [p 4]
0.3 What are the Linux benefits? [p 4]
0.4 I don’t believe in free software, etc. [p 6]
0.5 "There ain’t no such thing as a free lunch" [p 6]
0.6 I need high security. With commercial software, I can sue them if things go wrong. [p 7]
0.7 I need standards. Big software corporations (Microsoft) provide standards. [p 7]

---

0. For the Undecided (Linux Benefits)

0.1 Fundamentally, why Linux?

If you truly enjoy working with computers, Linux is the operating system of your dreams. It is more fun than any other computer operating system around. However, the reason why Linux is truly revolutionary is that it is Open Software. Our science and technology works owing to the free availability of information and peer review. Would you fly a plane that was based on proprietary science and unreviewed design, a plane at the internals of which nobody but the manufacturer could look? Then, why would you trust a closed, unreviewed, proprietary operating system? Linux is ideally suited for a mission-critical application.

Making horseshoes was once a closely guarded trade secret. Science and technology exploded 500 years ago thanks to sharing the knowledge by the means of printing. In the early days of printing, many of those who dared to share were assassinated for revealing "trade secrets". Linux is for the computer age what Gutenberg was for writing. Hopefully there will be no assassinations this time :-) .

0.2 Is Linux for me?

Only you can answer this question. Linux is a mature, powerful and extremally versatile UNIX-like operating system. The power and versatility come with a price--you may need to be computer-literate in order to set-up and maintain Linux. Linux is relatively easy to use once the operating system and applications are set up properly. So, your mother will also be able to use Linux, if you set up an easy graphical account for her and put the proper icons/menus on her GUI desktop. Linux is secure, so your mother will not be able to damage the system no matter how hard she tries--unless it's with a hammer :-) .

Linux is quite different than MS Windows, so do not expect that if you can get around MS Windows, Linux will be obvious to you. You may need to learn. On the other hand, if you come from UNIX, Linux will be easy for you. If you don't know much about computers or you don't enjoy them, chances are Linux administration is not for you. If you don't know your hardware, Linux installation may be a challenge for you.

0.2a Linux is difficult for newbies.

This may be true. But the question is: do you really want to learn it?? None of the Klimas family (that writes this guide) has any computing science background, yet we use Linux every day and we love it.

0.3 What are the Linux benefits?

Linux can give you:

o A modern and very stable multi-user, multitasking environment on your inexpensive PC hardware, at no (or almost no) monetary cost for the software. Linux is a rich and powerful platform--don't think of it as a "poor people" operating system. Out-of-box Linux has as much capability as MS

Windows NT with $5000 in software add-ons, is more stable, and requires less powerful hardware for comparable tasks.

o Unsurpassed computing power, portability, and flexibility. A Linux cluster recently (April 1999) beat a Cray supercomputer in a standard benchmark. Linux is VERY standard--it is essentially a POSIX compliant UNIX. (Yes, Linux is a best-of-the-breed UNIX. The word "UNIX" is not used in conjunction with Linux because "UNIX" is a registred trademark.) Linux is most popular on Intel-based PCs (price), but it runs very well on numerous other hardware platforms, from toy-like to mainframes.

o A truely great learning platform. If you are a parent, you should be really glad your daugher/son does Linux--s/he will surely learn something of lasting value. If you are a teacher, you should consider installation of Linux at your school.

o Excellent networking capability built into your operating system. You think you don't need a network? Once you try home networking, you will never be able to live without it! What about connecting the two or more computers that you have at home and sharing your hard drives, CDROM(s), sound card(s), modem, printer(s), etc.? What about browsing the net on two or more machines at the same time using a single Internet connection? What about playing a game with your son over your home network? Even your old 386 with Win3.11 may become useful again when connected to your Linux Pentium server and it is able to use your network resources. All necessary networking software comes with standard Linux, free, just setup is required. And it is not a second-gear shareware--it is exactly the same software that runs most of the Internet (the Apache software runs more than 50% of all Internet web servers and Sendmail touches some 70% of all e-mail). The pleasure of home networking is something I was able to discover only owing to Linux.

o Connectivity to Microsoft, Novel, and Apple proprietary networking. Reading/writing to your DOS/MS Windows and other disk formats. This includes "transparent" use of data stored on the MS Windows partition of your hard drive(s).

o Dozens of excellent and free, general-interest applications.

o Hundreds of specialized applications built by researchers around the world (astronomy, information technology, chemistry, physics, engineering, linguistics, biology, ...). In many fields, Linux seems like "the only" operating system in existence. The software in this category is typically not very easy to use, but if you want the power, it is the best software that humanity has in these areas. Doubtful? Have a look at: http://SAL.KachinaTech.COM/Z/2/index.shtml for examples.

o Thousands of free applets, tools, and smaller programs. "Small is beautiful" goes well with Linux philosophy.

o Scores of top-of-the line commercial programs including Word Perfect and all the big databases (e.g., Oracle, Sybase, but no Microsoft's). Many (most?) of these are offered free for developers and for personal use.

o State-of-art development platform with many best-of-the-kind programming languages and tools coming free with the operating system. Access to all the operating system source codes, if you require it, is also free.

o Freedom from viruses, software manufactures "features", invasion of privacy, forced upgrades, licensing and marketing schemes, high software prices, and pirating. How is this? Linux has no viruses because it is too secure an operating system for the viruses to spread with any degree of efficiency. The rest follows from the open-source and non-commercial nature of Linux.

o The operating platform that is guaranteed "here-to-stay". Since Linux is not owned, it cannot possibly be put out of business. The Linux General Public License (GPL) insures that development and support will be provided as long as there are Linux users.

o A platform which will technically develop at a rapid pace. This is insured by the modern, open-software development model which Linux implements: "build-on-the-back-of the-previous-developer" and "peer-review-your-code" (as opposed to the anachronistic closed-software model: "always-start-from-scratch" and "nobody-will-see-my-code"). Even if the current "Linux-hype" died out, Linux will develop as it did before the media hype started.

If you wanted to learn first-hand about the General Public License, check these famous GNU documents:
http://www.gnu.org/copyleft/gpl.html
http://www.gnu.org/gnu/linux-and-gnu.html
http://www.gnu.org/philosophy/categories.html#TheGNUsystem

The licence under which Linux is distributed is probably the most important part of it.

0.4 I don’t believe in free software, etc.

And do you believe in the Internet? The Internet and Linux share underlying ideas and have common roots. Do you remember the disbelief about the Internet a few years ago, the endless, seemingly unbeatable arguments that the free Internet cannot exist? "Who pays for that, anyway?"

The reality is simple. Cooperation and good will can benefit many at the same time: your gain is not my loss. Internet works fine and is expanding at a rapid pace. So does Linux.

0.5 "There ain’t no such thing as a free lunch"

Brett Bazant <bbazant@shaw.wave.ca> wrote in LinuxToday (http://linuxtoday.com/cgi-bin/showtb.pl?tbsn=12450&sn=5418):

> The economic paradigm which makes this true depends upon scarcity of resources. Software resources are only scarce because we all keep software proprietary and secret. But not Linux! When I give you my software, it may create opportunity cost for me, but I get to keep it even after I’ve given it to you. It is a free lunch only rivalled in history by the loaves and the fishes.

0.6 I need high security. With commercial software, I can sue them if things go wrong.

Don’t count on suing. Things go wrong on many MS Windows NT machines every day, and there are no damages awarded by courts. Read your MS Windows license agreement to find out that there is no guarantee whatsoever that ANYTHING will work. Trying to sue would be a waste of your money.

Linux also provides no guarantees, although it is far more secure than any version of MS Windows. If you are really security-sensitive , you can use high-security tools built by companies that rely on the availability of the source code to design and test their security features (e.g., Kryptokom in Germany provides high security firewalls). The "security in obscurity" implemented in MS Windows has repeatedly been demonstrated to be a naive approach.

Here is a quote from David Kastrup, Research Engineer, Bochum, Germany (after the "Internet Week", http://www.techweb.com/se/directlink.cgi?INW19990329S0050):

> "Risk aversion is what dictates you use Linux and other open products, rather than NT. The risks with NT are entirely out of your control, and there is nobody you could sue if anything goes wrong. Why people still believe the myth that Windows in any form offers any bit of accountability "more" than Linux remains a complete riddle to me."

0.7 I need standards. Big software corporations (Microsoft) provide standards.

Perhaps that’s what people would expect from large corporations, but the reality is rather different. Once, big companies loved inventing nuts that could be undone only by their own service shops. Did these nuts become standard? Hardly. They didn’t because there was no public benefit involved, and they couldn’t because they were patented. Luckily, now we have open and free standards for nuts.

An example from the computer field. The "standard" MS Word file format has changed numerous times over the recent years. This keeps happening probably for a good business reason: as soon as other companies "reverse-engineer" the current Word format, Microsoft changes it. There are even sub-formats (a MS "fast-save" anybody?). It is also completely closed--Microsoft does not publish the specifications. How can the user benefit from this in a longer term?

Xavier Basora (http://www.osopinion.com/Opinions/XavierBasora/XavierBasora47.html) writes:

> "... Microsoft’s standards are both proprietary and arbitrary- the stealth incompatibility of Office 97 file formats with older versions of Office or the subversion of Open standards like XML with proprietary extensions that require Internet Explorer 5, MS Active server and so on, are sober reminders of what the company does to a market."

To add to the confussion, companies typically do not "standardize" on file formats but on applications that are supposed to produce them. It is like standardizing on a manufacturer of nuts instead of nuts. How this is supposed to work if the file format keeps changing?

There are a few text/document oriented file formats that are quite definitely more standard than MS Word file format: ASCII, SGML, HTML, LaTeX, TEX, XML, PostScript, pdf, dvi ... and all of them have excellent support under Linux. The Word file format can be also read/written very well under Linux by Word Perfect, Star Office, Applixware, etc. to cover your today's needs.

The story is similar with other proprietary computing "standards". Linux, by its very nature, is based on true, published, and free standards because "open source" makes the full specifications available to everybody (competitors or not).

Go to Part 1: Before Linux Installation

LINUX NEWBIE ADMINISTRATOR GUIDE

ver. 0.56 1999-10-29

Part 1: Before Linux Installation

by Stan, Peter and Maria Klimas

Answers to Some Frequently Asked Linux Questions

Distributed under the General Public Licence http://www.gnu.org/copyleft/gpl.html. Your feedback, comments, corrections, and improvements are appreciated. Send them to bklimas@magma.ca

---

Quick site navigation:

Start: Linux Newbie Administrator Guide
Part 0: For the Undecided (Linux Benefits)
Part 1: >Before Linux Installation<
Part 2: Linux Resources, Help and Some Links
Part 3: Basic Operations FAQ
Part 4: Linux Newbie Administrator FAQ
Part 5: How to Upgrade the Kernel
Part 6: Linux Shortcuts and Commands
Part 7: Essential Linux applications (proprietary or not)

Contents of this page:

1. Before Linux Installation

1.1 Which Linux distribution should I use? [p 10]
1.2 What are the Linux hardware requirements? [p 11]
1.3 Will my hardware work under Linux? [p 13]
1.4 How do I download Linux? [p 13]
1.5 How do I get a Linux CD? [p 14]
1.6 I have RedHat CD but no install floppy. What do I do? [p 14]
1.7 What do I need to read before installation? [p 16]
1.8 Can I have MS Windows and Linux installed on the same computer? [p 16]
1.9 How to I partition my hard drive? [p 17]
1.10 The MS Windows partition occupies my whole harddrive. Can I shrink/split it without a re-install? [p 19]
1.11 How do I start the installation? [p 19]
1.12 Is the Linux installation difficult? [p 19]
1.13 I finished the installation. How do I login for the very first time? [p 20]

1. Before Linux Installation

1.1 Which Linux distribution should I use?

The differences between the distributions ("distros") are minor: the installation program, choice of the bundeled tools/applications, arrangement of a few things on the hard drive (most of Linux is still at the same, standard hard drive location in all distributions). Whichever distribution you decide to install, you will end up with essentially the same Linux.

We mostly use "Red Hat Linux" (also called RedHat or RH) for these reasons:
(1) RedHat is very popular (both advantage for a newbie and a testimony to RedHat quality).
(2) RedHat is a general-purpose distribution.
(3) RedHat comes with a relatively easy setup program.
(4) RedHat contributions to Linux are "open software" (this means that all the software written by RedHat and included on the RH distribution CD is lisensed under the General Public License, GPL, so that it can be legally copied, given away, reused, etc.).
(5) RedHat can be obtained very cheaply or free if you don't care for commercial support. This is a consequence of (4).

In short, as a newbie, you can safely bet on "Red Hat" unless you like something else or have specialized needs. The most recent RedHat release (October 1999) is version 6.1 (called RH6.1) which is only a relatively minor upgrade over the previous version of RedHat 6.0 (RH6.0) which hit the street in May 1999. The yet previous version was RedHat 5.2 and it was very popular for quite a while. Be sure to specify the most recent version if ordering your software from a dealer--many dealers like to clear their inventory by sending you an old version (this applies not only to Linux). The authors have no connection to RedHat (or any other Linux distributor) whatsoever.

Our recommending RedHat for newbies does not mean that other distributions don't offer benefits or unique features which may surpass RedHat in specific areas. We do believe that we benefited from an exposure to a different distribution because it helped us understand Linux better.

We tried Debian and we liked it very much. It was probably as easy as RedHat, but it seems less common (hence, being newbies, we picked up RedHat). The great benefit of Debian is that it is 100% non-commercial (put together by volunteer hackers, the true Linux way) and it probably most strictly adheres to Linux standards (it probably sets the standards too). Another benefit is that Debian crams on their numerous distribution CDs thousands of tools and applications--easily much more than any other distribution. All these tools/apps are nicely "packaged" (for ease of installation) and tested for compatibility. This makes Debian distro look monumental, safe, conservative, and always slightly outdated. So yes, we would not have a problem recommending Debian for as a great general-purpose Linux distribution.

Corel is currently beta-testing their own Linux distribution apparently geared towards a nice and easy platform to run the Corel suite of office applications: WordPerfect wordprocessor, QuattroPro spreadsheet, Corel Presentations, Paradox database, CorelDraw artist package.... The Corel Linux is based on Debian. So if you think about using Corel in the future, picking Debian may be your smart choice now.

Slackware seems to be favorite among "cutting-edge hackers" who like being close to the operating system and perhaps upgrade their kernel every other day--we did not use it so this is hearsay. S.u.S.E distribion is very popular in Europe. It surely looks German-solid, general-purpose distro with an easy setup and an excellent reputation. Another well-known and respected distribution is Caldera, said to be aiming at corporate users--it has the most fancy, perhaps most easy installation program. Mandrake distribution is a RedHat clone which is updated more often than the pedigree RedHat (RedHat tends to update less often so as not to proliferate too many versions). Mandrake seems to be very popular lately. There are "localized" versions of Linux for specific countries or languages (Korean, Chinese, Japanese ...)--they likely contain on default all the hacks and docs that the users in these countries probably want to see.

There are also "special purpose" distributions, e.g. the "real-time" editions of Linux (might be useful if you are in for automation, robotics, etc.), very small distros (could be good for the next-generation CD/MP3 player or if you like the idea of running Linux from a single floppy), parallel computing and clustering systems (might be great if you plan to do your own weather forcasting :-)), etc. Here the differences will be larger, but these distributions are not meant to be "general purpose". As a newbie, you likely don't want to start with any of these, although you might be tempted to. (They surely show Linux strength and viability--it runs on toys as well as computer clusters that make the currently fastest systems in the world.)

The distribution you need is of course specific to the hardware platform you have. This means that for your PC hardware containing a Intel 386 processor, or 486, or Pentium, or Cyrix, or K6, or similar, you need the binary distribution called "Intel" or "386". [Unless you are prepared to start with your own compilation of the Linux source code, which is not typical for a newbie :-)] This happens because there are binary distribution for other platforms too (PowerPC, Alpha, and perhaps a dozen more)--don't get those for your PC clone, surely they will not work.

In short, although newbies get confused with the multiple Linux distributions, there are reasons to have different distros. They should be viewed as a Linux strength rather than weakness.

This guide concentrates on RedHat 5.2, 6.0 and 6.1 on the PC (Intel) platform. Many of the answers will work fine on other distributions or platforms, but we did not try them.

1.2 What are the Linux hardware requirements?

"Out-of-box" Linux will run on a 386SX-based PC with 8 MB of memory, but such a low-end computer is practical for text-only applications (no X-window). A 486 with 16 MB memory and 600 MB free (unpartitioned) hard drives will do for work with X-windows. My 586-90 MHz with 64 MB of memory flies under Linux. My Pentium-2 233 MHz with 64 MB of memory is a complete pleasure with an almost instantaneous response even when running many large applications concurrently. My 486-33 MHz with 8 MB memory and 1 GB hard drive has too little memory to run adequately stand-alone under GUI, but is useful in my home network environment (a 486-class machine performs just adequately also stand-alone if it has at least 16 MB of memory). My old portable 386-SX-20 MHz Toshiba with 9 MB memory and 120 MB hard drive runs "legacy applications" under MS Windows 3.11 and connects to our Linux home network and is thus still useful. We tried Debian Linux on this Toshiba too, and it runs fine in text mode.

If you are willing to jump through a few extra hoops, you should be able to install and run Linux on a little as 4 MB of memory, but this is probably not worth the effort for the general purpose home Linux machine. I would say: get at least 16 MB of memory, and if possible 32 or 64 MB --more memory can make a difference in performance when running concurrently several large GUI applications. Memory is cheap these days.

Networking is where Linux really shines, so consider getting 10-base-T Ethernet cards--they are not very expensive and will be perfect to connect your two or more home computers together. Also, look around for old Ethernet cards which MS Windows deems obsolete--they can be bought for a really low price and they will work fine under Linux. To connect just two computers, a cross-over cable for direct Ethernet-card to Ethernet-card connection is sufficient. To connect more than 2 computers together, you need a hub (~US$30 to US$80) and normal (not cross-over) cables. (If you have extra Ethernet cards, you may also consider installing more than 1 Ethernet card on a computer, use direct connections using the cross-over cables, and save the expense of a hub.) The 10-base-T system uses "giant phone" (RJ45)-type connectors and all machines are connected to one box (called the hub). The hub has an extra connection (called "uplink") which you will use if you ever have a permanent connection to the outside world.

Here is another suggestion on setting up a different kind of network, older type, which uses coaxial cables. For this, no hub is necessary. Because this networking scheme is older, it can be assembled using cards and parts that are often available almost for free:

> (edited for space) From: John.Edwards@brunel.ac.uk Subject: Linux Guide-a suggestion
>
> Hi. Many older 10Mbps network cards (and some newer ones as well) have a BNC connector and you can usually pick up old co-axial cabling when companies upgrade to UTP. Add a T piece for each machine and a 50-ohm terminator at each end (about 1 pound or $1.50 each) and you have a home network that will happily support more machines than you probabily have room for. And most importantly--no expensive hub (or cheap hub that can cause trouble). There are other advantages to co-ax as well, it's tougher to break and more resistent to noise from other equipment.
>
> Disadvantages: There is a limit of 185 metres per network segment of thin co-ax, 30 machines per network, and you're stuck at 10Mbps, but I don't see any small home network needing more than that. Also if one cable goes down then the whole network stops, this shouldn't happen often unless someone unplugs a cable section. You can disconnect the T piece from a PC without harming the rest though.
>
> Quick diagram, T for a T piece and Term for a terminator:

```
Term-T-----T-----T-----T-Term
     |     |     |     |
     PC    PC    PC    PC
```

> The various parts connect together using BNC connectors similar to TV & video connector but with a bayonet that secures the two sockets together.

For more detail see page 6 of the /usr/doc/HOWTO/Ethernet-HOWTO

1.3 Will my hardware work under Linux?

Not every piece of PC hardware is supported under Linux, but most are, particularly the more standard, older, and popular ones. This applies to SCSI adapters, CDROMs, writable and rewritable CDs (CD-R and CD-RW), video cards, mice, printers, modems, network cards, scanners, Iomega drives, etc.

The most notable exceptions are the so-called Winmodems (=MS Windows modems). Avoid these like fire--they are maybe $5 less expensive than full modems, but they are crippled and there is little chance they will ever work under a normal operating system (they may not even work with the next version of MS Windows). External modems are never "MS Windows modems" so if in doubt, purchase an external modem (external modems are more expensive, but they don't drain your PC power supply, are easily portable between machines, look better, and show modem activity).

Another area of potential problem is the video card. If you have a recent "cutting edge" 3D or uncommon card, you may want to check its compatibility at http://www.Xfree86.org.

So the short answer is yes, in all likelihood your standard PC will run Linux with no problems. You don't invest much when trying Linux, so probably the easiest way to make sure is to attempt an installation on your existing hardware. There are Linux hardware compatibility lists at http://www.redhat.com/support/docs/hardware.html and http://metalab.unc.edu/LDP/HOWTO/Hardware-HOWTO.html if you want to check your newer or less popular hardware.

When purchasing new hardware, ask your supplier if the hardware is supported under Linux or consider a system with Linux pre-installed. Not too many major suppliers offer one yet (most plan to), but you can get one from a smaller vendor.

If a piece of hardware of yours is (apparently) not supported in your current Linux distribution, don't give up. Chances are that: 1. It is supported, but you don't know how to set it up. (Solution: stay around with Linux for a few days or weeks, don't waste your time, when you get some understanding of how your system works, then you may be able to set it up.) 2. You have to go through a more complex setup to support the hardware (for example some cryptic command or a kernel re-compile, which is not as difficult as it seems). 3. An updated (different?) distribution already supports it "out-of-box" (you can usually order it for US$1.99). 4. There is already an upgrade somewhere on the Internet, you have to find it, download it, and figure out how to install it. 4. The upgrade will be available next week--Linux development goes really fast!

1.4 How do I download Linux?

Do yourself a favor and do not download Linux, get an installation CD instead. Linux can be downloaded completely from the Internet, but it is a large and sophisticated operating system. The download will take hours or days of download time and you will probably encounter problems when trying to stitch the downloaded pieces together.

If you do have a speedy Internet connection (definitely not a 28.8-modem) and you are not a complete newbie, Linux download may be an option to you after all. Try: ftp://ftp.redhat.com/pub/ if the site is not too busy today.

1.5 How do I get a Linux CD?

Many possible ways. (1) Buy the "RedHat" CD from linuxmall (http://www.LinuxMall.com/), or cheapbytes (http://cart.cheapbytes.com/cgi-bin/cart)--last time I checked , "the unofficial" RedHat 6.1 GPL was US$1.89 and US$1.99 respectively + shipping and handling. They will mail you a bare CD. You get no printed manual, no support, no boot diskette, but the price is right, and the manual and tools to make a boot diskette are on the CD. I purchased several packages from "cheapbytes" and they always arrived fast, were of good quality, and there were no problems with my credit card charge (the authors have no connection to "cheapbytes" whatsoever). (2) Buy the boxed "official Red Hat" from the same place on the Internet or from your favorite software supplier, prices start at around US$40--you will get the printed manual, e-mail or telephone installation support (60 days?), the boot diskette, additional CDs with "bundled" commercial applications, and perhaps other goodies (free updates?). (3) Copy the installation CD from your friend. This is perfectly legal and ok--Linux is free. If you have a Linux CD, don't be shy to loan it to your neighbor. (4) Check your library, local bookstore, or http:/www.amazon.com. Several Linux handbooks come with an attached CD containing a full Linux distribution. This is a good way to start with Linux! It is definitely a good idea to have a nice Linux handbook. With Linux countless utilities, I need a handbook all the time. (5) Visit a Linux "installfest" when one is organized in a place near you. Local Linux "gurus" will install Linux on your computer free (bring the computer) and you will likely be able to get a Linux CD too (why don't you bring 1 or more empty CD-R to the fest?). Check for the Linux User Group on the net to see when the nearest to you plans an installfest. Good way to meet other Linuxers too. (6) If you are really short on cash or are a Linux pioneer in one of these places in the world where US$2 goes a long way, try "linux exchange" or "giveaway"--people loan or donate Linux CDs (e.g. try: http://visar.csustan.edu/giveaway.html). If you live in a major city, you may even get lucky and somebody will help you with your first Linux installation / setup. If you can't find anything, let me (bklimas@magma.ca) know. I'll try to arrange sending a copy of Linux CD to you (if you are not a joker). Once again, if you have a Linux CD, don't be shy to loan it to your neighbor.

1.6 I have RedHat CD but no install floppy. What do I do?

If your computer can boot from the CD drive (most computers cannot), you don't need a boot diskette to install Linux. Have a look at your BIOS setup, the boot sequence can often be set up there. My computer has the CD drive specified as the first boot device in the BIOS yet still cannot boot from the CD drive. So the BIOS setup does not necessary reflect the capability of your machine. If you can boot from CD drive, just insert the RedHat CD into the CD drive and reboot the computer to enter the RedHat Linux installation program.

If you don't know how to access your BIOS setup, read this paragraph. The BIOS setup can typically be entered at boottime by pressing the proper key at the right moment (often when a prompt is briefly displayed). Most often, it is the <Del> key. Here is a list of key combinations used by popular BIOSes: Acer notebooks: <F2> during Power-On Self-Test (POST). American Megatrends (AMI): <Del> during Power-On Self-Test. Award: <Del>, or <Ctrl><Alt><Esc>. Compaq: <F10> after the

square appears in the top right corner of the screen during boot-up. Dell: <Ctrl><Alt><Enter>. DTK: <Esc> during Power-On Self-Test. IBM Aptiva 535: <F1> while the square with the wavy lines is displayed in the upper right corner during power-on. IBM PS/2: <Ctrl><Alt><Del>, then <Ctrl><Alt><Ins> when the cursor is in the top right corner Mr. BIOS: <Ctrl><Alt><S> during Power-On Self-Test. Packard Bell: For some models, <F1> or <F2 > during Power-On Self-Test. Phoenix: <Ctrl><Alt><Esc> or <Ctrl><Alt><S>, or <Ctrl><Alt><Enter>.

If your computer cannot boot from the CD drive, make an install boot diskette from under DOS or the MS Windows DOS mode. (You have go to to "Shutdown" and "Restart in MS-DOS mode", not just run a DOS window). From DOS, you may try the program:

`EZSTART.BAT`

which is on my RedHat CD from "cheapbytes".

It is important that you have a perfectly good floppy (without even one bad cluster). The program that makes the diskette does not check if the floppy was written properly. Also don't count that the DOS `FORMAT` utility will find a faulty floppy--it probably won't. If I were you, I would make two or three boot floppies at once--you may be surprised how many diskettes have problems. For me, the third floppy worked! If your install diskette does not boot, make another one--it definitely should boot.

If you don't have the `EZSTART.BAT`, here are the essential commands. To make the boot floppy run:

```
F:\dosutils\rawrite.exe -f F:\images\boot.img -d a: -n
```

To make the supplemental (optional) diskette run:

```
F:\dosutils\rawrite.exe -f F:\images\supp.img -d a: -n
```

This assumes your CDROM is the DOS "F:" drive, and your floppy is "A:", adjust the commands if the drive letters are different on your system.

You may find it easier to run `rawrite` without any argument--it will interactively prompt you for the input image and output file.

From under Linux, you can make a boot disk by mounting the RedHat CDROM and typing the commands (as root user):

```
cd /mnt/cdrom/images/
dd if=boot.img of=/dev/fd0
```

[The ‘‘dd’’ command copies files. The above command specifies that the input file ("if") is ‘‘boot.img’’ and the output file ("of") is /dev/fd0, which is the first floppy drive, i.e. the floppy drive number zero (if you want to write to your second floppy drive, use /dev/fd1).]

1.7 What do I need to read before installation?

You may want to read the RedHat manual. If you don't have the printed copy, an html version is on your CD so you can read it using any web browser, e.g. Netscape for Windows. Look here to see how to access this manual and some additional reading material.

It is also helpful to get some UNIX orientation if you don't have any. Buy a good Linux manual or dust your old Unix handbook. Almost all basic Unix commands will run fine under Linux. Manuals for MS Windows are useless (click this, click that), but manuals for Linux/UNIX are typically great (give you an understanding of the system, a lasting benefit).

You may want to learn about your hardware: how many and what size hard drives you have, the type, number, order and size of all partitions on each drive, where are your DOS/Windows partitions, which one is the DOS/Windows boot partition (if you plan to have dual boot), what type of mouse you have, what video card and with how much memory, what monitor (max synchronization frequencies), etc.

Go to BIOS setup to see the number and geometry of your hard drives. Run DOS "fdisk" to display your hard drive(s) partition table(s). Watch your system boot to learn about the type of your video card and the amount of video memory. Boot MS Windows, go to the control panel-devices and write down the sound card, modem, network card types and settings (name, type, IRQ, i/o address, DMA channel). Read the label underneath your mouse to see the type of mouse you have. (Next time you buy a mouse, get a Linux-ready 3-button Logitech or similar--Linux makes good use of all three buttons.) Dust off your monitor manual to find out the maximum synchronization frequencies (vertical and horizontal) that your monitor supports. Never use freqencies out the the monitor specification--this may damage your monitor.

1.8 Can I have MS Windows and Linux installed on the same computer?

Yes, you can. Many Linuxers use dual boots. This is typically achieved by installing MS Windows on one hard drive partition and Linux on another partition. Linux comes with a simple boot manager called LILO, which will let you choose at the boot time the operating system you boot. Install MS Windows first and Linux only afterwards or else MS Windows installation program will disable your access to Linux. Have a Linux boot floppy ready if you need to re-install MS Windows--MS Windows will surely disable your access to Linux and you will have to boot Linux from the floppy and then re-run the command `lilo` to get back to business.

From under Linux, you will be able to read from and write to your MS Windows drive partitions so that data exchange between MS Windows- and Linux-based program is seamless. You will also be able to use your existing MS Windows-based resources: sound files, backgrounds, pictures, fonts, etc. (First check if it does not violate your license agreement :-) though. For products that are on rent to you from Microsoft, it probably would. With my Linux computers, I am proud to have no pirated software on my system whatsoever.)

1.9 How do I partition my hard drive?

If you plan a dual boot (Linux and MS Windows on the same computer), first use your DOS/Win utility `FDISK` to make the MS Windows partition(s). Leave a part of the hard drive(s) unpartitioned for Linux. You will make and format the Linux partitions during your RedHat installation. Make the MS Windows partition "primary" and "bootable". Install, configure, and test your MS Windows before Linux installation. If you plan to run Linux only, you need just clean hard drive (no partitions) to start with.

It is possible to have only one Linux partition (plus one for MS Windows if you dual-boot). But it is better to have more partitions so you can keep users' data separate from the rest of the operating system. This way, if something ever goes wrong and you have to reformat and re-install the operating system, you don't lose the users' data. (You can however do a Linux re-install without losing the contents of the `/home` directory that contains all user data if you skip the "re-format" option given to you during installation.)

During the Linux setup, you will be asked to partition the available space on your hard drive(s). There are many possible ways to partition, depending on your hard drive space, requirements, and taste. I like Linux hard drive partitions like this (for a modest total of 2 GB that I dedicated for Linux):

```
mount point    type     size
/              ext2     300 MB
/usr           ext2     1200 MB
/home          ext2     380 MB
swap           swap     120 MB
```

In the above example, I dedicate 300 MB for the root partition that holds the base of the Linux operating system. I give 1200 MB for the mount point that will be visible on my filesystem as the `/usr` directory and will contain user's programs (the programs that don't come with the base operating system and I install later, for example WordPerfect). I dedicate 380 MB for the partition that will be visible as the directory `/home` and will contain the setting and data of all users on the machine. And I give 120 MB for a "raw" partition for the operating system to use as the virtual memory (extension of the physical, silicon memory on the hard drive, so-called swap). If your kernel is lower than 2.2 (this is the case with standard RH5.2 and earlier), your swap partition cannot be larger than approximately 127 MB. The rule of thumb is that the swap should be about twice the amount of the physical memory (RAM). 120 MB is quite a bit of swap and it is unlikely you should need more. If you think you do need more (e.g. you expect to run custom programs with really large data structures) you might want to create a larger swap partion during the installation (or several smaller swap partitions) or add swap a file(s) later.

2 GB is a lot of disk space and should be sufficient even for users who like having many applications. (This is because Linux applications tend to be slimmer than their MS Windows equivalents). However, if you try to install everything that's available for Linux on the Internet, you will surely run out of disk space :-) . My experience is that however large the hard drive space, it will get filled and I regret I don't have more :-) .

If my space on the hard drive is really restricted, I may consider a two-partition setup like this (for a lean 650 MB total dedicated for Linux):

```
mount point    type     size
/              ext2     600 MB
swap           swap     50 MB
```

In this example, I dedicate 600 MB to contain the base of the operating system, applications, and user documents/data, and give 50 MB for the swap partition (for the operating system to use as the virtual memory). The 50-MB swap should be quite sufficient for medium duty operations. The limitation of 600 MB for the operating system, applications and user data means that you will have to be selective as to which applications you install or else you risk running out of hard drive space. Try pressing <F1> when installing the optional software that come on Red Hat CD--it will give you a short description what the software does so you could perhaps decide if you really need it. (Don't worry too much if you miss something you need, you can install the missing parts later). You can easily finish the RedHat installation with 200 MB free on your Linux partition (out of 600 MB used in this example) if you make reasonable choices. Please note that "bundling together" the root partition "/" and the /home directory will likely save you some disk space, but it is not the safest solution.

It is possible to install Linux on even less disk space than in the example above, but you will have to be even more selective as to what you install.

For a larger available hard drive space, I may consider the following setup (for a comfortable total of 6 GB dedicated for Linux):

```
mount point    type     size
/              ext2     400 MB
/usr           ext2     2000 MB
/usr/local     ext2     2500 MB
/home          ext2     800 MB
swap           swap     300 MB
```

Please note that the the mount points can reside on different physical hard drives.

Another consideration when setting up the partitions. Many older BIOSes have the restriction that the boot partition cannot extend beyond the 1024th cylinder on your first physical hard drive. To overcome this limitation, simply make the first (bootable) partition so that it ends before the cylinder number1023 (this makes this partition max approximately 512 MB in size, which is plenty for the "/" root partition). Once Linux boots, the BIOS restriction does not matter any more as Linux can also access the partition(s) beyond the cylinder number1023.

When installing and using Linux, your drives appear as devices with the following names: hda--first IDE drive (stands for "hard drive a", i.e. the master drive on the first IDE interface), hdb--second IDE drive (i.e., the slave drive on the first IDE interface), hdc--third IDE drive (i.e. the master drive on the second IDE interface), hdd--fourth IDE drive (i.e. the slave drive on the second IDE interface). The numbers mean the partitions on the physical drives: "hda1" means the first IDE hard drive (hd a), first partition (1); "hda2" is the first IDE hard drive, second partition; "hda3"--the first IDE hard drive, third partition; (and so on if you have more than 3 partitions on the first IDE hard

drive); "hdb1"--second IDE hard drive, first partition (or just "hdb" if it is the CDROM installed as a slave on your first IDE interface). "hdc1"--third IDE hard drive, first partition, etc. SCSI drives have analogous names but start with letters "sd" (="SCSI drive"), followed by the letter indicating the SCSI interface and by the number indicating the SCSI device id. For example, "sda4" means "first SCSI interface, id number 4". If you have an external zip drive attached to your parallel port, it will apear as a SCSI device "sda4" (it works in a SCSI-emulation mode).

1.10 The MS Windows partition occupies my whole harddrive. Can I shrink/split it without a re-install?

Possibly. There is a utility called `FIPS.EXE` on your RedHat CD that does just that. Check the directory `\dosutils\fipsdocs\`on your RedHat CD for documentation. If I were you, I would back-up my essential data before doing anything to the partition. There are also commercial utilities to change the partition size without destroying the content of the partition.

My personal preference is to do a clean re-install of MS Windows on a single, dedicated partition. I leave some space on the hard drive unpartitioned so I can use it later for Linux. My fair division of hard drive space between MS Windows and Linux is 50/50. Linux programs tend to be smaller but they include as a standard components that MS Windows offers only with many thousand dollars of add-ons: e.g., servers (not just clients) for telnet, ftp, http, and mail, several databases, programming lanuages, ...

1.11 How do I start the installation?

Insert the installation boot diskette into your floppy drive, the RedHat CD to the CDROM, and reboot. If you boot from the CD, insert the RedHat CD into your CDROM drive and reboot.

You may also start the installation from DOS (or real DOS mode under MS Windows), by running `EZSTART.BAT` which is on my RedHat CD from Cheapbytes.

1.12 Is the Linux installation difficult?

It was not for me. It seems that for most newbies, it is fairly straight forward and painless. Depending on your hardware and installation choices, it typically takes 0.5 h to 2.0 hours. [Expect longer or possible problems for slow systems with very restricted memory--it took whole night to install RH on my 486-33 MHz with 8 MB memory, the system pausing for 5 minutes at a time appearing to do nothing, yet it installed ok.] However, some newbies reported that the installation was a "total nightmare" to them (hardware problems?). If you encounter problems, my advice would be to install a plain-vanilla system, without struggling with the highest resolution on your fancy video card or other bleeding-edge hardware which you might have. Anything can be added/configured later, after you get more understanding of how things work on your system. Even a re-install is always an option for a newbie (it seems Linux gurus think it is a shame to ever re-install). It seems that many newbies have problems because they specify too high screen resolutions (which may be not supported or supported only with some extra tune-up). Again, it may not be wise to break your whole installation for support of a single device--the support can be added/tuned-up later.

1.13 I finished the installation. How do I log-in for the very first time?

As root. "root" is the only account that exists after the initial installation (newer installation programs do prompt you to create a regular user account during the installation). Example login:

```
my_machine_name login:  root
Password:  my_password
```

In the example above, I typed the word "root" at the login prompt. After that, I entered the password that I chose during the initial Linux installation. The password did not appear on the screen when I typed it (for security).

"root" is a special account with an absolute power over the system, and it is used for system administration. You surely want to create at least one more "user" account later to perform regular (not system administration) work. Read on to learn how to do it.

Go to part 2: Linux Resources, Help and Some Links

LINUX NEWBIE ADMINISTRATOR GUIDE

ver. 0.56 1999-10-29

Part 2: Linux Resources, Help and Some Links

by Peter, Stan, and Maria Klimas

Answers to Some Frequently Asked Linux Questions

Distributed under the General Public Licence http://www.gnu.org/copyleft/gpl.html. Your feedback, comments, corrections, and improvements are appreciated. Send them to bklimas@magma.ca

---

Quick site navigation:

Start: Linux Newbie Administrator Guide
Part 0: For the Undecided (Linux Benefits)
Part 1: Before Linux Installation
Part 2: >Linux Resources, Help and Some Links<
Part 3: Basic Operations FAQ
Part 4: Linux Newbie Administrator FAQ
Part 5: How to Upgrade the Kernel
Part 6: Linux Shortcuts and Commands
Part 7: Essential Linux applications (proprietary or not)

Contents of this page:

2. Linux Resources, Help and Some Links

2.1 Any Linux reading materials? [p 21]
2.2 Is there a help command? [p 22]
2.3 Newsgroups [p 23]
2.4 Any Linux Internet links? [p 24]

---

2. Linux Resources, Help and Some Links

2.1 Any Linux reading materials?

Check your RedHat CD for the softcopy of the RedHat manual (written with newbies in mind). Under Linux, you can read it using:

```
lynx  /mnt/cdrom/doc/rhmanual/manual/index.htm
```

This will start lynx, a simple text-mode HTML browser to view the manual. You can also use Netscape or any other html browser. If your RedHat is still not installed, you can view the manual with an MS Windows-based browser, e.g. Netscape for Windows. Just access the file `index.htm` on your CD in the directory as shown in the command above.

If you already installed Linux, check `/usr/doc/LDP` for the excellent Linux Documentation Project manuals. (If you didn't install the documentation, consider installing everything now.) For example, these commands will let you browse the Linux System Administrators' Guide:

```
cd /usr/doc/LDP/sag
lynx sag.html
```

Also, check `/usr/doc/HOWTO` for the HOWTO documents, and `/usr/doc/HOWTO/mini` for the MINIHOWTOs. Check `/usr/doc/FAQ` for a set of FAQ. For example, these commands will let you view the text version of the Linux-FAQ on your cdrom (the CD must be mounted first):

```
cd /mnt/cdrom/doc/FAQ/txt/
less Linux-FAQ
```

Since the documentation is also available on the RedHat CD, so you can read it before Linux installation. For example, this will let you read the Linux-FAQ from under DOS, assuming your CDROM under DOS is the drive D:

```
d:
cd \doc\FAQ\txt
edit Linux-FAQ
```

For more or updated documentation, see http://metalab.unc.edu/mdw/index.html#guide.

2.2 Is there a help command?

Most Linux commands can be run with the "--help" option. For example, this will give you a concise help on the Linux `cp` (copy) command:

```
cp --help | less
```

More extensive info is accessed from the command line using the so-called manual pages `man` *`topic`* . For example:

```
man cp
```

will display the manual page for the "cp" command. The manual pages are the standard "help" system under Linux, and contain wealth of detailed, technical information, but often require an effort to understand by a newbie.

The `man` command uses the simple utility called `less` that lets you scroll through a text. Use arrows to scroll, press "q" to quit. (Actually, `less` can do more than this. Press "h" for help when running `less` , or learn more about `less` using `man less` .)

There is also the info command `info topic` . For example:

```
info cp
```

Often `info` contains information similar to `man`, but more up-to-date. Unfortunately, the `info` navigating utility is not very intuitive.

If you don't remember exactly the name of the command that you need to use, try `apropos`. For example, to obtain a list of commands which have something to do with "copy", try:

```
apropos copy
```

In some menu driven programs, most notebly under `setup` when browsing the system services, you may press F1 for info what the particular service does.

The kde windows manager includes a GUI-based "help browser", which can be started by clicking the "book with a light" icon on the Kpanel. This browser can be used to access the kde-specific help as well as the system manual pages. The Gnome desktop contains a similar help system.

If you want to learn about the many packages that come on your CDs in rpm format, you may want to use the GUI-based glint (type `glint` in an X-terminal) to browse through the packages, display the info that they contain, and install them if you wish (the installation has to be done as root). Under RH6.0, the `glint` program is substituted with `gnorpm` , which is much faster, but also somewhat less convenient to use.

2.3 Newsgroups

These can be a intimidating place to be--the world's strangest wackos seem to be all browsing the newsgroups. I just choose to ignore the stupid or offensive postings or e-mails. For the malicious ones, I make an exception and inform the system administrator at their originating e-mail provider. Advertisments which I receive after posting to a newsgroup get deleted before reading--I know I am not the only one doing this, so please mark your subject line clearly if you want your e-mail to be read, particularly if your e-mail address contains the string "aol". Despite their drawbacks, newsgroups can be an efficient way of accessing the information you need.

Before going to the newsgroups, I would highly recommend DejaNews (http://www.dejanews.com/home_ps.shtml). This is a huge archive of newsgroup postings and you can search it using nice search tools. This way, you can often find an answer to your question without going through tons of trash, and without exposing yourself to anger after posting a question which "was already asked ten times this week". You may be surprised by the amount of information avaiable through Dejanews.

There are several newsgroups devoted to Linux and they seem much better than other newsgroups (maybe they are better policed by the Linux expert?). Here is a short list:
news:comp.os.linux.announce (moderated--the postings are done by a moderator, who reviews them prior to the posting. Inspect the footer of any message for info how to post.)
news:comp.os.linux.help
news:comp.os.linux.setup

news:comp.os.linux.misc (miscellenous)
news:comp.os.linux.advocacy (Use this one for discussion of pros and cons of Linux and perhaps a comparison of Linux with other operating systems. This is an excellent newsgroup if you like getting into endless arguments).
news:alt.linux.sux (Here you can read/write really all opinions on Linux.)
news:comp.os.linux.networking
news:comp.os.linux.x (X-windows)
news:comp.os.unix (general UNIX newsgroup)

Please note that there is a newsgroup etiquette ("netiquette"), and you risk rejection and perhaps expose yourself to flames if you choose to break it. The major points:
- don’t post on a topic that is unrelated to the subject of the newsgroup;
- don’t post to many newsgroups at the same time (cross-post);
- use plain ASCII, don’t post attachments, pictures, html, etc.;
- don’t advertise (particularily commercial products).

Useful abbreviations/acronyms:

BTW	"By the way, "
AFAIK	"As far as I know," (S/he may acctually know the best, but is being cautious)
RTFD	"Read the ’fine’ documentation!" (Impatient and probably rude)
ASAP	"as soon as possible"
FYI	"For your information," (Implying that everybody else knows about it. S/he cannot mean that it is confidential if s/he puts it on a newsgroup, can s/he?)
IMHO	"In my humble opinion," (Very polite.)
:-)	A smile to you. Many variants are available, also showing other faces to express moods.
c.o.l.a.	The newsgroup com.os.linux.announce (or perhaps comp.os.linux.advocacy?)
Tia	"Thanks in advance!"
iirc	"If I remember correctly,"
fwiw	"For what it’s worth" (The responder feels that the anwser may not be a definitive one.)

Please mail me if you know of other interesting abbreviations that a newbie Linux administrator ought to know.

2.4 Any Linux Internet links?

There must be thousands of Internet sites devoted to Linux. Here are some Linux links which I like, in no particular order. If you need something else, you should find a useful pointer on one of these pages.

http://sunsite.auc.dk/linux-newbie/	Master site for this document (LNAG). Bookmark it.
http://www.kalug.lug.net/linux-admin-FAQ/	Linux Admin FAQ (the non-Newbie).
http://members.aa.net/~swear/pedia/learning-linux.html	Gary's Encyclopedia--Learning Linux. Bookmark it.
http://jgo.local.net/LinuxGuide/	Josh homepage. Good resource for learning Linux.
http://www.control-escape.com/	This site seems good for newbies!
http://qlink.queensu.ca/~3srf/linux-admin/	Linux administration made easy (LAME). "All new and improved" (Apr. 99). Recommended.
http://metalab.unc.edu/mdw/index.html#guide	Lots of Linux documentation. Bookmark it.
http://www.slashdot.org/	Discussions for nerd, hackers, gurus, etc.
http://www.freshmeat.org/	Update on today's releases of Linux software
http://linuxtoday.com/	Linux news--excellent daily reading. Bookmark it.
http://www.llp.fu-berlin.de/	The Linux Lab Project. Data acquisition etc.
http://www.linuxberg.com/	Linuxberg. Big portal. They have everything there. I like their rating of Linux software and am installing only packages that received 5 penguins ;-) . Bookmark it.
http://counter.li.org/linuxcounter_eng.html	The Linux counter. Register yourself as a linuxer!
http://www.ap.univie.ac.at/users/havlik/Album/Linux-Counter/	See Dennis Havlik's impressive maps on Linux growth and geographical distribution.
http://www.cl.cam.ac.uk/users/iwj10/linux-faq/index.html	Linux FAQ.
ftp://sunsite.unc.edu/pub/Linux/	Tons of Linux software at the Sunsite archive. Bookmark it.
http://stommel.tamu.edu/~baum/linuxlist/linuxlist/linuxlist.html	Linux applications.
http://www.boutell.com/lsm/	Linux applications.
http://www.linuxlinks.com/Software/	Linux applications.
http://directorysearch.mozilla.org/Computers/Operating_Systems/Linux/	Great new portal (better than yahoo) with excellent links for Linux newbies.
http://dir.yahoo.com/.../Unix/Linux/	Yahoo's entries for Linux.
http://www.debian.org/	Debian Linux site.
http://www.redhat.com/support/docs/hardware.html	Linux hardware compatibility list.
http://metalab.unc.edu/mdw/links.html	Lots of useful Linux links
http://www.cse.unsw.edu.au/~conradp/linux/	Scores of execellent links.
http://www.linuxstart.com/documentation/	More links to Linux documentation.
http://www.linuxlinks.com/	Even more Linux links.
http://www.gnu.org/	Master GNU site (GNU's-Not-Unix. This is a recursive definition).
http://www.redhat.com/	The Red Hat site. It is typically too busy to bother.

Go to part 3: Basic Operations FAQ

LINUX NEWBIE ADMINISTRATOR GUIDE

ver. 0.59 1999-11-05

Part 3: Basic Operations FAQ

by Stan and Peter Klimas

Easy Answers to Questions Frequently Asked by the New Linux Users who Install Linux on their Home Computers or Administer their Home Network

Distributed under the General Public License http://www.gnu.org/copyleft/gpl.html. Your feedback, comments, corrections, and improvements are appreciated. Send them to bklimas@magma.ca

---

Quick site navigation:

Start: Linux Newbie Administrator Guide
Part 0: For the Undecided (Linux Benefits)
Part 1: Before Linux Installation
Part 2: Linux Resources, Help and Some Links
Part 3: >Basic Operations FAQ<
Part 4: Linux Newbie Administrator FAQ
Part 5: How to Upgrade the Kernel
Part 6: Linux Shortcuts and Commands
Part 7: Essential Linux applications (proprietary or not)

Contents of this page:

3.1 Basics

3.1.1 Filenames [p 28]
3.1.2 What are the different directories for? [p 30]
3.1.3 How do I run a program? [p 31]
3.1.4 How can I change the PATH? [p 33]
3.1.5 How can I shutdown my computer? [p 34]
3.1.6 How do I deal with a hanged program? [p 35]

3.2 Users, passwords, file permissions, and security

3.2.1 Home directories, root, adding user [p 38]
3.2.2 About password security [p 39]
3.2.3 I forgot the root password [p 39]
3.2.4 I have file permission problems. How do file ownership and permissions work? [p 41]
3.2.5 My mp3 player chokes. The sound is kind of interrupted (how to set suid) [p 45]

3.3 Job scheduling with "at", "batch", and cron

3.3.1 How do I execute a command at a specified time (using "at"or "batch")? [p 46]
3.3.2 How do I set up cron? [p 47]

3.4 Swap space [p 48]

3.4.1 Swap partitions [p 49]
3.4.2 Swap files [p 49]

3.5 Shell

3.5.1 What's a shell and do I want to use a different one? [p 50]
3.5.2 How do I write a simple shell script? [p 51]

3.6 Simple programming

3.6.1 How do I write a simple Python program? [p 52]
3.6.2 How do I write a simple GUI program (using tcl)? [p 52]
3.6.3 How do I write a simple perl script? [p 53]
3.6.4 How do I compile a simple C program? [p 54]

3.7 How do I install a program I downloaded from the Internet? [p 54]

Basic Operations FAQ

3.1 Basics

3.1.1 Filenames

Linux is case-sensitive. For example: my_filE, my_file, and my_FILE are three different files. Your password and login name are also case-sensitive. (This follows the tradition since both UNIX and the "c" programming language are case-sensitive.) Naming conventions for files and directories are identical.

Filenames under Linux can be up to 256 characters long and they normally contain letters, numbers, "." (dots), "_" (underscores) and "-" (dashes). Other characters are possible but not recommended. In particular, it is not recommended to use special metacharacters: "*" (asterisk), "?" (question mark), " " (space), "$" (dollar sign), "&" (ampersand), any brackets, etc. This is because matacharacters have special meaning to the Linux shell. It is possible to have space inside the filename, but we don't recommend it either--we use underscore instead.

It is not possible at all to have '/' (slash) as a part of the filename because '/' is used to represent the top of the directory tree, and as a separator in the pathnames (the same as '\' is in DOS).

To manipulate files with names that do contain matacharacters, I use a pair of ’ (apostrophes), so that the metacharacters are quoted and therefore the shell does not interpret their meaning. For example, to remove a file `my file*` (contains space and asterisk), I would issue:

```
rm ’my file*’
```

Please note that I use a pair of ’ (apostophes) for quoting. Quoting with a pair of " (quotation marks) is generally weaker than quoting with ’ . When you use " some metacharacters may get interpreted.

Here is the meaning of some metacharacters:
* = Matches any sequence of zero or more characters (except for "." dot at the beginning of a filename).
? = Matches any single character.
[abC1] = Matches a single character in the enumerated set. In this example the set is: ’a’, ’b’, ’C’, or ’1’.
[a-z] = Matches any lower-case letter.
[A-F] = Matches any upper-case letter from A to F in the Latin alphabet.
[0-9] = Matches any single digit.
[a-zA-Z0-9] = Matches any letter (lower or upper case) or any digit.

Examples. This command will show any filename in the current directory, with the exception of filenames starting with "." (dot):

```
ls *
```

An equivalent to this command is to type just "ls" or "dir". Files with names starting with "." are not shown
because "." as the first character of a fileneme is not matched by "*". Think of files with names starting with "." as an equivalent of DOS hidden files. Use `ls -a` (list with the option "all") or `ls .*` to see these "dot" files. The dot-files are common in the user home directories and are typically used to hold user-level configurations.

This command will show any filename that contains a dot:

```
ls *.*
```

This command will show any filename that contains two dots:

```
ls *.*.*
```

Please note that Linux does not have "filename extensions" and some other DOS-like file-naming features ("Micros~1.doc" comes to mind).

This command will show all filenames in the current directory that start with "a" or "b", or any capital letter:

```
ls [abA-Z]*
```

As an example of problems that you might face when using non-recommended characters in a filename, try creating a file with a name starting with a dash and then remove it--there seems to be no way to do it (a dash normally introduces command options). E.g., the command
`dir > -junk`

will create such a funny file (the symbol ">" redirects the output from the `dir` command to a file named `-junk`). Since the regular way of removing the file `-junk` does not work, I use:

```
rm ./-junk
```

The dot slash at the beginning means "the current directory" and here just serves the purpose of hiding the leading dash so it is not interpreted as introducing an option to the `rm` command. The point here is that I rather stick to traditional naming conventions than face the occasional complications.

3.1.2 What are the different directories for?

This is explained very nicely in the Linux System Administrator Guide (SAG), which should be available on your system. Try:

```
cd /usr/doc/LDP/sag
lynx sag.html
```

This will start a text-mode browser "lynx" to view this html book. You can also use any other browser, e.g. Netscape for Windows, to view this book. This book and other LDP books are actually quite easy to read.

Briefly, Linux contains five filesystems. These filesystems can reside on a single or different physical hard drives and/or hard drive partitions, depending on the size and need of your system. (A single filesystem can also be distributed between different physical devices, if needed.)

The root `"/"` filesystem--contains basic operating system and maintenance tools. The content of this filesystem should be sufficient to start up the system and perform emergency maintenance and repairs if they were necessary.
`/usr` filesystem--contains all commands, libraries, documentation, and other files that do not change during normal operation. This will also contain major applications, perhaps the ones that come with your distribution, for example Netscape.
`/var` filesystem--contains files that change: spool directories, log files, lock files, temporary files, and formatted manual pages.
`/home` filesystem--contains user files (users' own settings, customization files, documents, data, mail, caches, etc).
`/proc` filesystem--contains entirely illusionary files. They don't really exist on the disk and don't take any space there (although `ls -l` will show their size). When viewing them, you really access information stored in memory. It is used to access information about the system.

The parts of the **root filesystem** are:
`/bin`--commands needed during bootup that might be used by normal users.
`/sbin`--commands not intended for use by general users (users may still use them).
`/etc`--system-wide configuration files for your operating system.
`/root`--the home directory of the system administrator (called super-user or root).
`/dev`--device files. Devices appear on Linux as files so it is easy to write to them.
`/mnt`--mount points for removable media (floppy, cdrom, zipdrive), partitions of other operating systems (like dos), network shares, and anything else that is mounted on the file system temporarily. It normally contains subdirectories for the mounting shares.
`/lib`--shared libraries for programs that reside on the root filesystem.
`/boot`--files used by LILO (a bootstrap loader, the thing that loads first when the computer is booted and perhaps gives you an option which operating system to boot, if you have more than one OS on your computer). It typically also contains the Linux kernel, but this can be stored somewhere else, if only LILO is configured to know where it is.
`/opt`--optional large applications, for example kde under RedHat 5.2 (under RedHat 6.0, kde is distributed as any other X-windows distribution, main executables are in the `/usr/bin` directory).
`/tmp`--temporary files. This directory may clean automatically.
`/lost+found`--files recovered during the filesystem repair.

The most interesting parts of the **/usr filesystem** are:
`/usr/X11R6`--X-windows system.
`/usr/X11`--the same as /usr/X11R6 (it is a symbolic link to /usr/X11R6).
`/usr/X11R6/bin` --lots of small X-windows apps, and perhaps symbolic links to the executables of some larger X-windows applications that reside in other subdirectories).
`/usr/doc`--Linux documentation.
`/usr/bin` and `/usr/sbin`--similar to their equivalents on the root filesystem (`/bin` and `/sbin`), but not needed for basic bootup (e.g. during emergency maintenance).
`/usr/local`--the installed "local user" applications, for example Netscape (each application in a separate subdirectory).
`/usr/local/bin`--perhaps smaller "user" apps, and symbolic links to the larger executables contained in separate subdirectories under `/usr/local`.

It is important to understand that all directories appear in a single directory tree, even if the directories are contained on different partitions, physical drives (including floppies, etc), or even if they distributed over the network. Therefore, there are no DOS-type "drive letters" under Linux.

3.1.3 How do I run a program?

Typing the name of the executable on the command line doesn't help? There are three possibilities.

The first possibility: you don't type the name of the executable correctly. Check the case--Linux is case sensitive! For example, typing "Pico" or "PICO" will not start the `pico` editor.

The second possibility: maybe the program is not on your PATH. Under Linux (or UNIX), an executable must be on your PATH to run it, and the current directory is NOT on your PATH. Type the full path to the executable with the executable name, or execute:

```
cd the_program_dir
./program_name
```

You must put the dot and slash in front of the program name or the program will NOT execute. (This is a security feature not to put one's current directory on the path. It makes "trojan horses" more difficult. A "trojan horse" is a malicious program that pretends to be something different than it really is.) This dot means "the current directory", and the slash "/" is a separator between the directory name and the filename (exactly as "\" in DOS).

You may check your path using:

```
echo $PATH
```

To learn how to change your PATH, or add your current directory to it, see the next answer [p 33] .

If your executable is lost somewhere in your directory tree, you may want to find it using (for example):

```
find -name "netscape"
```

to find a file called "netscape". You may be able to achieve the same result faster using:

```
locate netscape
```

(Locate runs faster because it relies on a pre-built database of files on your system. This database if updated by a background cron process that normally runs at night, so don't count on `locate` to find a file if you regularily switch off your computer for the night, or you search for a file that you just installed.)

Please note that the PATH is normally different for root than regular users (root's PATH includes `/sbin` and `/usr/sbin` whereas users' don't). Therefore users cannot execute command located in the "sbin" directories unless they specify the full path to the command. Also, if you become a superuser by executing the `su` command, you inherit the user's PATH, and to execute the command located in sbin, you need to specify the full path.

Conversly, if you need to learn where an executable which is on your PATH is located on your system (i.e., the executable runs by typing its name anywhere in the system, but you would like to know where it is), you may use something like this:

```
which netscape
```

which will show a full PATH to the executable program called "netscape".

The third possibility: maybe the file is not executable. If it should be, change the permissions to make it executable. E.g. (as root or the user who owns the file):

```
chmod a+x my_file
```

will make the file "my_file" executable for all users. Check if it worked using:

```
ls -l  my_file
```

Read here [p 41] if you don’t understand the output of this command or the whole "third possiblity".

3.1.4 How can I change the PATH?

The PATH is the list of directories which are searched for the program the execution of which you request. You can check your PATH using this command:

```
echo $PATH
```

which, on my system , shows the PATH for the user "yogin" to be:

```
/opt/kde/bin:/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/home/yogin/bin
```

The ":" is a separator, therefore the above PATH represents a list of directories as follows:

```
/opt/kde/bin
/usr/local/bin
/bin
/usr/bin
/usr/X11R6/bin
/home/yogin/bin
```

Here is the output from the command "echo $PATH" run on my system on the account "root":

```
/opt/kde/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin:/root/bin
```

You can change the PATH for all users on the system by editing the file `/etc/profile` and adjusting (as root) the line starting with "PATH=". I do it using the `pico` editor (as root):

```
pico -w /etc/profile
```

(The option -w turns off the wrap of long lines.)

Re-login for the change to take effect. To set up the PATH for an individual user only, edit the file `/home/user_login_name/.bash_profile` (please note the dot in front of the filename--files starting with a dot are normally invisible, you have to use `ls -a` to see them).

If you really want to have the current directory on your PATH, add "." (dot) to your PATH. When used in the place when directory name is expected, a dot means "the current directory". The specification for the path in `/etc/.bash_profile` may then look like this:

```
PATH="$PATH:$HOME/bin:."
export PATH
```

This command takes the contents of the environmental variable called PATH (as set for all users in `/etc/profile`), and appends to it the name of your home directory as set by the variable HOME with an attached "/bin" and then a dot. Finally, the command assigns the resulting string back to the variable called PATH. It is necessary to use the command "export" after modifying PATH or any other user-enviroment variable, so that the variable is visible outside of the script that sets it.

3.1.5 How can I shutdown my computer?

In a text terminal, press <Ctrl><Alt><Del>, wait for the shutdown process to complete, and turn off your machine only after it starts rebooting again. If you are in X-windows, first switch to a text terminal by pressing <Ctr><Alt><F1> (three keys simultanously). Do not turn off your machine without the proper shutdown or else you may have disk error messages next time you boot. (Typically, the errors resulting from improper shutdown will be repaired automatically during the next boot, but occassionally more serious problem may result, and then you may need to repair the files manually or re-install!)

If you prefer your computer to go to the halt after you press <Ctrl><Alt><Del> (instead of the default reboot), you can set this up by editing the file /etc/inittab. This file specifies something like this:

```
# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

Change (as root) the option "-r" to "-h" so that it reads:

```
# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -h now
```

The line starting with "#" is just a comment (it is for the humans, it does not have any effect on the computer).

Root can also use the `shutdown` command. This command can be used for either a local or remote shutdown of the system. The last one is very useful if a program hangs so that the keyboard is no longer functional. For example:

`telnet name_of_machine_with_no_operable_keyboard`
[login as a user]
`su`
[give password]

Now either execute `ps axu |more,` find the process id of the offending command in the ps output and do
`kill` *`pid_of_offending_process,`* or reboot your machine with:

```
/sbin/shutdown -rn now
```

This command will shutdown really fast, bypassing standard (longer) shutdown procedure--useful when the system becomes really buggy (the option -n will make "shutdown" kill all the processes before rebooting).

Please note that for security reasons, you cannot login to a remote machine as root (e.g., over the telnet). You have to login as a user and then execute `su` and give a password to become a super user (root).

The shutdown command may also be used to execute a shutdown later. E.g. (as root):

```
/sbin/shutdown -r 23:59
```

will reboot the system 1 minute before midnight.

If the `shutdown` command is too long for you, you may want to try these two commands, which do exactly what their names suggest (as root):

```
reboot
halt
```

A fancy way to shut down your computer is to switch your system to the runlevel 0 (for halt) or runlevel 6 (for reboot). Try it using (as root):

```
init 0
```

The meaning of the different runlevels is explained in the file `/etc/inittab` and here.

3.1.6 How do I deal with a hanged program?

Buggy programs do hang under Linux. A crash of an application should not, however, affect the operating system itself so it should not be too often that you have to reboot your computer. In our experience, a misbehaving operating system may be a sign of hardware or configuration problems: we repeatedly encountered problems with the Pentium processor overheating (the fan on the Pentium did not turn as fast as it should or it stopped altogether, the heat sink on the Pentium was plugged with dirt), bad memory chips, different timing of different memory chips (you may try re-arranging the order of the chips, it might help), wrong BIOS setup (you should probably turn off all the "advanced" options, Linux takes care of things by itself). The "signal 11" error message is typically associated with hardware problems and it most likely to manifest itself when you perform computing-intensive tasks: Linux setup, kernel compilation, etc.

Not really hanged. Some programs might give the uninitiated impression of hanging, although in reality they just wait for user input. Typically, this happens if a program expects an input file name as a command line argument and no input file is given by user, so the program defaults to the standard input (which is console). For example, this command

```
cat
```

manipulate the file `/etc/shadow` as you would `/etc/passwd` .

To make the "floppy access" to my system a little bit more difficult, I considered running a computer without a floppy drive :-) Unfortunately, Linux CDs are bootable these days. I set up my boot sequence (in the BIOS setup) so that the system boot from the hard drive before floppy and CDROM are tried, and added an "administrative" password on changes to the BIOS settings. Still, I worry that these BIOS passwords are so easily crackable and also one could remove my harddrive and connect it to another computer for reading :-) . I am considering an "encrypted file system" which is now available on Linux, but considering all the trouble associated with it, perhaps I will settle on locking my room :-) . If all this sound paranoid to you, it probably is--it just illustrates the point there is little computer security, even under Linux, if the potential cracker has physcial access to your hardware.

3.2.3a I forgot my user password.

If a regular (non-root) user forgets his/her password, this is not a problem since root can change any password. For example (as root):

```
passwd barbara
```

will prompt for a new password for the user "barbara" (no knowledge of old password required). If a user (non-root) wants to change his/her password, s/he will be asked for the old password first. (This is a security feature so nobody changes your password if you left your terminal unattended.)

If you need to temporarily disable any user account, there is no need to change his/her password. Just put an asterisk "*" at the begining of the password field (before the encryped password) in the file `/etc/passwd` or `/etc/shadow` . The "*" means that no login is permitted for this account. When you want to restore the account, you just erase the star and the user account is back in operation, with its old password.

3.2.4 I have file permission problems. How do file ownership and permissions work?

Linux (the same as UNIX) is a secure, multiuser operating system, and this creates a level a complexity with "files permissions". Trouble with file permissions can lead to unexpected and nasty problems. Understanding file permissions is of uttermost importance to be able to administer any multiuser operating system (be it UNIX, WinNT, or Linux). My advice would be: learn the system of Linux (or any UNIX) file permission conventions, you will not regret it.

The permission conventions are the same for normal files and directories, so whatever is said about files below, applies also to directories. It is also important to remember that lower level directories (and the files the directories contain) inherit the permissions of their parents. Therefore don’t expect to have any permissions to the subdirectory (or a file residing in it) if you don’t have any to the parent.

File owners. Each file belongs to an owner (typically a login name) and to a group. The owner is typically the person who created (or copied) the file. The group often consists of one person--the owner, and has the name identical to that of the owner, but it does not need to be so. A file can be

removed (erased) only the owner of the file, or a member of the group that owns the file, or the root. Yet other users may be able to modify or erase the content of the file if they are given the permission to do so--read on. The owner and group that owns the file will be shown in the output from the `ls -l` command (="list in the long format"). For example, the command:

```
ls -l junk
```

produced this output on my screen:

```
-rwx------   1 yogin    inca           27 Apr 24 14:12 junk
```

This shows the file "junk", belonging to the owner "yogin" and to the group "inca".

The ownership of a file can be changed using the commands `chown` (change owner) and `chgrp` (change group), which are normally executed by root:

```
chown peter junk
chgrp peter junk
ls -l junk
```

After executing the above 3 lines, the command `ls-l junk` produces this output on my screen:

```
-rwx------   1 peter    peter           27 Apr 25 20:27 junk
```

Changing the file ownership comes handy if you move/copy files around as root for use by other users. At the end of your housekeeping you typically want to hand-in the file ownership to the proper user.

File permissions. Now, an owner of a file can make the file accessible in three modes: read (r), write (w) and execute (x) to three classes of users: owner (u), members of the group (g), others on the system (o). You can check the current access permissions using:

```
ls -l filename
```

If the file is accessible to all users (owner, group, others) in all three modes (read, write, execute) it will show:

```
-rwxrwxrwx
```

Skip the first "-" (it shows "d" for directories, "-" for normal files, "l" for links, "c" for character devices, "b" for block devices). After this initial character, the first triplet shows the file permission for the owner of the file, the second triplet whows the permissions for the group that owns the file, the third triplet shows the permissions for other users. A "no" permission is shown as "-". Here is an output from the `ls -l` command on a file that is owned by root, for which the owner (root) has all permissions, but the group and other can only read and execute:

```
drwxr-xr-x   2 root     root        21504 Apr 24 19:27 dev
```

The first letter "d" shows that the file is actually a directory.

You can change the permissions on the file which you own using the command chmod (="change mode"). For example, this command will add the permission to read the file "junk" to all (=user+group+others):

```
chmod a+r junk
```

In the command above, instead of "a" (="all"), I could have used "u", "g" or "o" (="user", "group" or "others"). Instead of "+" (="add the permission"), I could have used "-" or "=" ("remove the permission" or "set the permission"). Instead of "r" (="read permission"), I could have used "w" or "x" ("write permission" or "execute permission").

For example, this command will remove the permission to execute the file "junk" from others:

```
chmod o-x junk
```

Instead of letters, one can also use numbers to specify the permissions. To understand how it works look at this:
read=4
write=2
exectute=1

The total permission for a class of users is the sum of the three. Thus:
1 = execute only (seems unusual)
2 = write only (seems unusual)
3 = write and execute (seems unusual)
4 = read only (common)
5 = read and execute (common)
6 = read and write (common)
7 = read, write and execute (common).

The permission for all the three classes of users (owner, group, others) is obtained by gluing the three digits together one by one. For example, the command

```
chmod 770 junk
```

will give the owner and his/her group the completto of permissions, but no permissions to others. The command:

```
chmod 666 junk
```

gives all three classes of users (owner, group, others) the permissions to read and write (but not execute) the example file named "junk". Please note the "666". It is quite often used and, at least for one person I know, it is a proof that Linux (any UNIX for that matter) is a work of the devil >:-0.

This command:

```
chmod 411 junk
```

would give the owner the permission to read only, and the group and others to execute only. This one does not seem useful, but might be funny, at least for those North American Linux users who dial 411 (telephone number) for directory assistance. Mail me if you can think of any other funny permissions (maybe 007?).

The numerical way of representing file permissions is called "octal" because the numbers have base 8 (the decimal system's base is 10). The highest digit in the octal system is 7 (octal system has eight digits: 0 to 7, analogous to decimal system having ten digits: 0 to 9). The octal representation is really a convenient notation for the binary represention of file permissions, where each permission is flagged as "set" or "denied" with a one or zero and the total is represented as a string of zeroes and ones, as in this diagram:

```
user class:                                  owner  group others
example permissions:                          rwx    rw-    r--
absent permissions:                           ---    --x    -wx
binary representation of the permissions:     111    110    100
octal representation  of the binary:            7      6      4
```

Default file permissions with umask. When a new file is created, it is given default permissions. On my system, these are:

```
-rw-r--r--
```

This means that files created by a user can be read and written to be the user, but only read by his/her group and others. Still, on my default RH5.2 system, users cannot read the files in the other users' home directories because the permissions on the home directories is:

```
drwx------
```

I can check the default file permissions given to my newly created files using:

```
umask -S
```

(The option "-S" stands for "symbolic" and tells `umask` to display the permissions in a easy-to-read form.)

I can change the default file permissions for newly created files using a command like:

```
umask u=rwx,g=,o=
```

which will give the owner all the permissions on newly created files, and no permission to the group and others.

Using numbers to set default permissions with umask is more tricky. The number shows the permissions that you take away for users (opposite to `chmod`). Thus:

```
umask 000
```

will give full permissions to everybody on newly created files. The next example gives full permissions to the owner, none for everybody else (perhaps that's what one may want):

```
umask 077
```

3.2.5 My mp3 player chokes. The sound is kind of interrupted (how to set suid).

The MP3 player might not be given enough processor power (it requires a lot of it). It could be that your system is lousy. Or you might be running too many cpu-intensive programs at the same time. Or, most likely, you may need to run the player with a higher priority. (The priority of a program can be set with the command `nice` -- see `man nice`or`info nice`). Try to run the player as root--programs run by root are given higher priority than those run by normal users. If this solves the problem, set the "suid" on the executable so all users are given the "effective user id" of root when running it, for example:

```
chmod a+s x11amp
```

will do the trick for the xamplifier program. The output from

```
ls -l x11amp
```

on my computer is now:

```
-rwsr-sr-x   1 root     root       319172 Mar 13  1998 x11amp
```

The first "s" indicates that the substitute-user-id (suid) bit is set. The second "s" indicates that the substitute-group-id (sgid) is also set. Thus anybody who executes `x11amp` is given the effective user id of the program owner and effective group id of the owner group, which in the example above is the user "root" and the group "root".

Setting the suid for a program could possibly become a security hole in your system. This is unlikely the case on a closed home network and when setting suid for a program origin of which is well traceable. However, even at home, I wouldn't suid a piece of code origin of which is uncertain, even if the setup instructions urged me to do so.

Some programs do however require suid for proper functioning, for example `kppp` (the popular modem "ppp" conection utility under the KDE graphical-user-interface desktop).

3.3 Job scheduling with "at", "batch", and cron

3.3.1 How do I execute a command at specified time (using "at" or "batch")?

The `at` command will execute the command(s) you specify at the date and time of your choice. For example, I could start playing music from my CDROM at 7 o'clock in the morning:

```
at 7:00
cdplay<Ctrl>d
```

In the example above, I entered the first line "at 7:00" on the command line and then pressed ENTER. To this, the `at` command displayed a prompt "at>". At this prompt, I entered my command "cdplay" and then pressed the control key and "d" simultaneously to finish the input. If instead of pressing <Ctrl>d , I pressed "ENTER", the next "at>" prompt would appear, at which I would be able to enter the next command to be executed right after "cdplay", also at 7:00. And so on, I could have had many commands scheduled for execution one by one starting at 7:00. After typing the last command, I would finish the input with <Ctrl>d. Think of the <Ctrl>d as sending "end-of-file" to the current input. Don't press <Ctrl>d twice because this will log you out--that's what <Ctrl>d does when entered straight on the Linux command line.

You can list the job you scheduled for execution using:

```
at -l
```

which will give you the numbered list of the jobs waiting.

If you changed your mind, you can remove a job from this list. For example:

```
atrm 8
```

will remove the job with the number eight on the list.

I could also schedule a job for execution much later, for example:

```
at 23:55 12/31/00
startx
```

would start my X-windowing system right on time for the new millennium (5 minutes before midnight on 31 of December 2000).

If you cannot execute the `at` command, check if the at daemon ("atd") is loaded (as root, use setup-"system services"). If you cannot execute the `at` command as a regular user although it works for root, check if the empty file `/etc/at.deny` exists and there is no file `/etc/at.allow`. This should be the default setup and it permits all the users to execute `at`. If you want only certain users to use `at`, create a file `/etc/at.allow` and list these users there.

For other options, check:

```
man at
```

If you wish to perform a processor-intensive job in a background when the system load is low, you may choose to use the `batch` command. For example, I could run setiathome (a program cranching data to help in search of extraterrestial intelligence, SETI) using:

```
batch
at>setiathome<Ctrl>d
```

In this example, I entered the command `batch` and then, at the "at>" prompt, I entered the command which I wanted to be executed in the background. The job tries to start immediately, but goes ahead only when the system load is under 0.8 You can check the system load by inspecting the contents of the (virtual) file `/proc/loadavg` . For example:

```
cat /proc/loadavg
```

When a batch job finishes, the output is sent to me via e-mail.

3.3.2 How do I set up cron?

Cron (a Linux process that performs background work, often at night) is set up by default on your RedHat system. So you don't have to do anything about it unless you would like to add some tasks to be performed on your system on a regular basis or change the time at which cron performs its duties.

Please note that some of the cron work might be essential for your system functioning properly over a long period of time. Among other things cron may:
- rebuild the database of files which is used when you search for files with the `locate` command,
- clean the /tmp directory,
- rebuilding the manual pages,
- "rotate" the log files, i.e. discard the oldest log files, rename the intermediate logs, and create new logs,
- perform some other checkups, e.g. adding fonts that you recently copied to your system.

Therefore, it may not be the best idea to always switch your Linux machine off for the night--in such a case cron will never have a chance to do its job. If you do like switching off your computer for nights, you may want to adjust cron so it performs its duties at some other time.

To find out when cron wakes up to perform its duties, have a look at the file `/etc/crontab`, for example:

```
cat /etc/crontab
```

It may contain something like this:

```
# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

You can see that there are four categories of cron jobs: performed hourly, daily, weekly and monthly. You can modify those or add your own category. Here is how it works.

The columns in the entries show: minute (0-59), hour (0-23), day of month (1-31), month of year (1-12), day of week (0-6--Sunday to Saturday). The "*" means "any valid value".

Thus, in the example quoted, the hourly jobs are performed every time the computer clock shows "and one minute", which happens every hour, at one minute past the hour. The daily jobs are performed every time the clock shows 2 minutes past 4 o'clock, which happens once a day. The weekly jobs are performed at 22 minutes past four o'clock in the morning on Sundays. The monthly jobs are performed 42 minutes past four o'clock on the first day of every month. The directory with the script file that contain the command(s) to be executed is shown as the last entry on each line.

If you wanted your jobs to be performed at noon instead of 4 in the morning, just change the 4s to 12s. Cron wakes up every minute and examines if the `/etc/crontab` has changed so there is no need to re-start anything after you make your changes.

If you wanted to add a job to your cron, place a script which runs your job (or a link to your script) in the directory `/etc/cron.hourly` or `cron.daily` or `/etc/cron.weekly`, or `/etc/cron.monthly`.

3.4 Swap space

Swap is an extension of the physical memory of the computer. Most likely, you created a swap partition during the initial RedHat setup. You can verify the amount of swap space available on your system using:

```
cat /proc/meminfo
```

The general recommmendation is that one should have: at least 4 MB swap space, at least 32 MB total (physical+swap) memory for a system running command-line-only, at least 64 MB of total (physical+swap) memory for a system running X-windows, and swap space at least 1.5 times the amount of the physical memory on the system.

If this is too complicated, you might want to have a swap twice as large as your physical (silicon) memory, but not less than 64 MB.

If you ever need to change your swap, here are some basics.

3.4.1 Swap partitions

You can have several swap partitions, each up to approximately 124 MB. Here are the steps to create and enable a swap partition:

- Create the partition of the proper size using `fdisk` (partition type 82, "Linux swap").

- Format the partition checking for bad blocks, for example:

```
mkswap -c /dev/hda4
```

You have to substitute /dev/hda4 with your partition name. Since I did not specify the partition size, it will be automatically detected.

- Enable the swap, for example:

```
swapon /dev/hda4
```

To have the swap enabled automatically at bootup, you have to include the appropriate entry into your /etc/fstab file, for example:

```
/dev/hda4 swap swap defaults 0 0
```

If you ever need to disable the swap, you can do it with:

```
swapoff /dev/hda4
```

3.4.2 Swap files

Swapping to files is usually slower than swapping to a raw partition, so this is not the recommended permanent swapping technique. Creating a swap file, however, can be a quick fix if you temporarily need more swap space. You can have up to 8 swap files, each with size of up to 16 MB. Here are the steps for making a swap file:

- Create a file with the size of your swap file:

```
dd if=/dev/zero of=/swapfile bs=1024 count=8192
```

This physically creates the swap file `/swapfile`, the block size is 1024 B, it contains 8192 blocks, the total size is about 8 MB. (The `dd` command copies files. In the example above, the input file (if) was `/dev/zero`, the output file (of) was `/swapfile`. You cannot use the `cp` (copy) command for creating a swap file because the swap file must be physically continuous on the hard drive.)

- Set up the file with the command:

```
mkswap /swapfile 8192
```

- Force writing the buffer cache to disk by issuing the command:

```
sync
```

- Enable the swap with the command:

```
swapon /swapfile
```

When you are done using the swap file, you can turn it off and remove:

```
swapoff /swapfile
rm /swapfile
```

You may also want to see the nice info written by Linus Torvalds himself:

```
man mkswap
```

3.5 Shell

3.5.1 What is a shell and do I want to use a different one?

A shell is the program that interprets what you type on the command line and decides what to do with it. A shell can also be invoked in a non-interactive way, for example to execute a pre-typed list of commands contained in a text file (a "shell script"). Think of a shell as an equivalent of the DOS "command.com" (command-line interpreter) and the shell script files as an equivalent of the DOS batch files (*.bat).

There are several shells available on the Linux system (if you installed them): `bash` ("Bourne Again" shell), `sh` (Bourne shell), `csh` (C shell, with a syntax akin to the "c" programming language), `pdksh` (public domain Korn shell), `tcsh` (enhanced C shell), `ash`, `zsh`, and perhaps a couple more.

The default shell on my system (and most probably on yours too) is `bash` , which is an excellent and standard shell, and I really cannot see a reason why a newbie like myself would want to change it. `bash` is fully backwards-compatibile with the Bourne shell (the most popular shell on UNIX) and incorporates many enhancements and best features from other shells. From a newbie prospective, the different shells are included with Linux for historical reasons and backwards-compatibility of shell scripts that may require a particular shell to run.

If you wanted to try another shell, type, for example

```
tcsh
```

which will start the enhanced c shell. When done, type

```
exit
```

which will return you to the previous shell.

The shell for each user is specified as the last field in the in the password file `/etc/passwd`. If you really wanted to change it, edit (as root) this file and replace the "/bin/bash" with the shell of your choice.

3.5.2 How do I write a simple shell script?

Create a text (ASCII) file that is to contain the shell script. For example, I would use the pico editor to write a script that runs the program `tar` with all the parameters usually necessary to uncompress a tarball downloaded from the Internet (I never seem to remember the `tar` options). I decided to call my script "untar":

```
pico untar
```

Since the file "untar" did not exist in my current directory, it was created by the `pico` text editor. Now, I type in the content of my script:

```
#!/bin/bash
echo this is the script file $0
echo untarring the file $1
# this calls tar with options -xvzf (extract, verbose, filter
through gzip, input filename)
tar -xvzf $1
```

I save the file with <Ctrl>o and exit with <Ctrl>x

The first line of the script, starting with "#!" (called pound-bang), is special--it tells the shell what program should be used to interpret the script. In this example, the script is to be interpreted by the bash shell `/bin/bash` . The first line must start with #! or the script will never run (the file will be interpreted as just a text file). Other lines starting with # are comments for the author (readers, users) of the shell and are totally ignored by the computer.

The $0, $1, $2 ... in my script are the parameters passed to my script. For example, if I ran a script called "myscript" with seven parameters like this:

```
myscript a b c d e f g
```

then $0 would be seen inside "myscript" as having the value "myscript", $1 would have the value "a", $2 would be "b", $3 would be "c", etc.

On the second and third line of my example script, the command `echo` prints on the screen everything that follows on the same line, expanding $0 and $1 to the values of the parameters passed to the script. The fourth line is a comment written to myself to remaind me what I was trying to achieve, just in case I ever had to modify my script. The fifth line performs the actual work.

Once the script is written, I make the text file executable to the file owner:

```
chmod u+x untar
```

and my script is ready to use:

```
./untar my_tar.tar.gz
```

Linux scripting is definitely rich, flexible, powerful and can be complex. However, it does not require special knowledge to write simple scripts for automation of common tasks. You just put togheter a group of often used commands, one by one, into a file. I use scripting because I am too lazy to type the same groups of commands over and over again.

3.6 Simple programming

3.6.1 How do I write a simple Python program?

Quite like writing a shell script. Create/edit a text file that will contain your Python program:

```
pico try_python
```

Type in some simple python code to see if it works:

```
#!/usr/bin/env python
print 2+2
```

The first line tells the shell how to execute this text file. The second is a simple python expression.

After saving the file, I make it executable:

```
chmod a+x try_python
```

after which I can run it by typing:

```
./try_python
```

Python is an excellent, and very modern programming language, so give it a try. Very good (and large) free handbooks by G. van Rossum (the Python creator) are available on the net (try: http://www.python.org/doc/ for browsing or ftp://ftp.python.org for downloading).

3.6.2 How do I write a simple GUI program (using Tk)?

Tk is a GUI extension of the easy yet powerful tcl programming language. For example, I may use `pico` to create a text file that will contain a simple tk program:

```
pico try_tk
```

and type in a simple example of tk code to see if it works:

```
#!/usr/bin/wish
button .my_button -text "Hello World" -command exit
pack .my_button
```

The first line (starting with the "#!" pound-bang) tells the shell what utility to use to execute my text file. The next two lines are an example of a simple tk program. First, I created a button called "my_button" and placed it at the root of my class hierarchy (the dot in front of "my_button"). To the button, I tied the text "Hello World" and a command that exists the program (when the button is pressed). Last line makes my program's window adjust its size to just big enough to contain my button.

After saving the file, I make it executable:

```
chmod a+x try_tk
```

after which I can run it by typing (in the X-terminal, because it requires X-windows to run):

```
./try_tk
```

Tk is a very popular for building GUI front ends.

3.6.3 How do I write a simple perl script?

Perl is a scripting language famous for its power, flexibility and perhaps cryptic syntax. It is also very popular amoung linux and unix (and not only) gurus. I may use `pico` (or any other text editor of my choice) to create a simple perl script. The script below does nothing useful, yet illustrates some features of perl:

```
#!/usr/bin/perl -w
#sample perl program
# name of scalar variables start with $
$a=2;
$b=3;
print $a**$b,"\n";
$hello_world='Hello World';
print $hello_world,"\n";
system "ls";
```

The first line tells the shell how to execute my text file. The option "-w" causes perl to print some additional warnings, etc. that may be useful for debugging. The next two lines (starting with #) are comments. The following lines are almost self explanatory: I assign some values to two variables ($a and $b), put $a to power $b and print the result. The "\n" prints a new line, just like in the "c" programming language. Then I define another variable to contain the string "Hello World" and, in the next line, I print it to the screen. Finally, I execute the local operating system command "ls", which on Linux prints the listing of the current directory content. Really stupid script.

Surely, I must remember to save the file, then make it executable (see the previous answers) and then I am ready to run it.

3.6.4 How do I compile a simple C program?

Start your favourite text editor and type in your source code. For example, I may use `pico`:

```
pico hello.c
```

and type in the Kerningham and Richie (the creators of "c") intro example of a C program:

```
#include <stdio.h>
void main(void) {
printf("hello world\n");
}
```

I save the file and then envoke the GNU C complier to compile the file "hello.c":

```
gcc hello.c
```

The gcc complier produces an executable binary file "a.out", which I can run:

```
./a.out
```

3.7 How do I install a program I downloaded from the Internet?

o If the program is a RedHat binary package (*.rpm), I use the RedHat package manager `rpm` . First I read the info on the package content:

```
rpm -qpi my_new_file.rpm
```

This queries (mode "q", must be the first letter after the dash) the yet uninstalled package (option "p") so that it displays the info (option "i") which the package contains. If I want to install the program, I run (as root):

```
rpm -ihv my_new_file.rpm
```

This runs `rpm` telling it to install the package (mode "i", must be the first letter after the dash) while printing to the screen more information than usual (option "h"=display "hashes" to show the unpacking progress, option "v" = be verbose). The contents of the package is distributed to the directories where it belongs. After this installation, the program is ready to run, I just have to find where the executable is. If I have trouble finding the executable, this lists all the files that the package contains together with their destination directories:

```
rpm -qpl my_new_file.rpm
```

This queries (option "q") the yet uninstalled package (option"p) so that it displays the listing (option "l") of all the files the package contains.

There are also GUI front-ends to `rpm`: glint (very slow, comes with RH5.2), gnopro (confusing, comes with RH6.0), and kpackage (by far the best of the three but it has to be installed separately--check your favorite Linux software site to download it).

o If what you downloaded from the net is a Linux source code in a form of a compressed tarball, the procedure is longer than with the binary-only rpm. I typically install the program as root.

First, I change my current working directory to `/usr/local` :

```
cd /usr/local
```

Second, I decompress the tarball that I downloaded from the net:

```
tar -xvzf
/home/the_directory_where_the_tarball_is/my_tarball.tar.gz
```

This extracts (option "x") the contents of the *.tar.gz (or *.tgz) tarball , unzips it (option "z") while talking to me more than usual (option "v" = verbose). Please note that the option "f" means "file", so the filename must immediately follow the letter "f". The contents of the tarball is extracted into a subdirectory which `tar` creates under my current working directory, which in this case is `/usr/local/` . The tarball knows what the new subdirectory should be called.

If the tarball is not compressed, I may use:

```
tar -xvf /home/the_directory_where_the_tarball_is/my_tarball.tar
```

Third, I have a look if the new subdirectory was created and how it is called, then I `cd` into it:

```
cd the_new_program_subdir
```

Since some of the directories have long names, I use the great autocompletion option to save on typing--I just type the first few letters and then press <TAB> .

Fourth, most programs are compiled by executing these three commands:

```
./configure
make
make install
```

The above commands can take some time to complete (0.5 h?). If any of them fail, it might be an idea to read the README or INSTALL or whatever info is provided with the new program. Some programs may require customization of the environment (e.g. addition of their directory to the PATH) or installation of an additional library, or yet something else. It can sometimes be a pain. Very simple programs might not need the "./configure" or/and "make install" step, in which case "make" alone will do.

Fifth, if everything goes well, I find the executable in the new program. The names of executables display in green when running this command:

```
ls --color
```

Now, I can run the executable:

```
./the_executable
```

Sixth, if I plan to run the program more often, I create a symbolic link to the executable from the directory `/usr/local/bin` :

```
cd /usr/local/bin
ln -s /usr/local/the_new_program_subdir/the_executable .
```

This way, the executable (actually, a symbolic link to it) is on my PATH and it can be run by simply typing its name (no need to type the full path to the executable any more). Some programs will put the executable (or a link to it) in a bin directory so you can skip the last step.

o There are also programs distributed as "source code rpm" packages. They require installation of the *.rpm package with the "rpm" utility and then compliation of the source code by executing the same: "./configure - make - make install" sequence as for the sourcecode distributed as tarballs.

Go to Part 4: Linux Newbie Administrator FAQ

LINUX NEWBIE ADMINISTRATOR GUIDE

ver. 0.57 1999-11-01

Part 4: Linux Newbie Administrator FAQ

by Stan, Peter and Maria Klimas

Easy Answers to the Questions Frequently Asked by the New Linux Users who Install Linux on their Home Computers or Administer their Home Network

Distributed under the General Public License http://www.gnu.org/copyleft/gpl.html. Your feedback, comments, corrections, and improvements are appreciated. Send them to bklimas@magma.ca

---

Quick site navigation:

Start: Linux Newbie Administrator Guide
Part 0: For the Undecided (Linux Benefits)
Part 1: Before Linux Installation
Part 2: Linux Resources, Help and Some Links
Part 3: Basic Operations FAQ
Part 4: >Linux Newbie Administrator FAQ<
Part 5: How to Upgrade the Kernel
Part 6: Linux Shortcuts and Commands
Part 7: Essential Linux applications (proprietary or not)

Contents of this page:

4.1 LILO issues

4.1.1 Linux cannot detect all my memory [p 58]
4.1.2 LILO displays only LI (or LIL) and hangs [p 59]
4.1.3 How can I change the operating system that LILO boots on default? [p 61]
4.1.4 The LILO prompt stays too short (or too long) on the screen during the bootup [p 61]
4.1.5 Uninstalling Linux [p 61]

4.2 Accessing my drives

4.2.1 Where are my drives? [p 61]
4.2.2 How can I access my CDROM? [p 62]
4.2.3 How to mount a floppy, zip drive, dos partition, or a network drive? [p 63]
4.2.4 How to mount a remote MS Windows filesystem through Samba? [p 65]
4.2.5 Any quick way to access a file on a DOS/Windows floppy? [p 65]
4.2.6 Mounting works when I am root. Can a normal user mount? [p 66]
4.2.7 Mounting command is too long, how can I simplify it with an alias? [p 68]
4.2.8 Can I automount? [p 68]
4.2.9 How do I get my zip drive recognized? [p 69]

4.2.10 Can I set 32-bit hard drive IO? [p 71]

4.3 Working with X-windows

4.3.1 How do I install kde on RedHat 5.2? [p 72]
4.3.2 How do I set up my video card, monitor and mouse for the X-server? [p 74]
4.3.3 How to X-window remotely? [p 74]
4.3.4 Can I have multiple sessions of X running at the same time? [p 76]
4.3.5 Can I have a GUI login prompt? [p 76]

4.5 Printer and soundcard

4.4.1 How to setup my soundcard? [p 77]
4.4.2 How do I setup my printer? [p 78]
4.4.3 Word Perfect 8 does not have a driver for my printer [p 80]

4.6 Setting up a network

4.5.1 Would it be worth it to set up my home network? [p 80]
4.5.2 How to set up my home network? [p 80]
4.5.3 I have problems configuring my ppp dial out [p 83]
4.5.4 How to browse the net from my networked computer without a modem? [p 85]
4.5.5 How do I use Samba? [p 86]
4.5.6 Sendmail [p 88]
4.5.7 Simple web server (running Apache) [p 89]
4.5.8 Simple ftp server [p 90]
4.5.9 How can one access my computer from the outside world when I am on the net using my phone ppp conection? [p 91]
4.5.10 Can my home computer get hacked? [p 92]

4.1 LILO issues

4.1.1 Linux cannot detect all my memory

If you have more than 64 megabytes of physical memory, Linux kernel ver. 2.0.36 or lower will use, by default, only the first 64 MB. To see how much memory Linux uses on your system, type:

```
cat /proc/meminfo
```

You can check your version of Linux kernel with:

```
uname -a
```

The last popular kernel with the "memory problem", 2.0.36, comes with RedHat 5.2. My RedHat 6.0 came with kernel 2.2.5-15 so it does not have the "memory problem" any more.

To get more than 64 MB memory recognized on RH5.2, you have to edit (as root) the file `/etc/lilo.conf`, and add a line like this just before your first "image=" statement:

```
append="mem=80M"
```

If you have an amount of memory different than 80 MB, adjust the above line. For any changes in `/etc/lilo.conf` to take effect, you *must* re-run the program

```
lilo
```

(watch if it runs without errors) and reboot. After the reboot, you can check if your adjustment worked using either of these two commands:

```
cat /proc/meminfo
free
```

For testing purposes, or if you are having problems, the option of specifying the amount of memory at the LILO prompt is useful:

[type at LILO prompt during bootup] `linux "mem=16M"`

Occasionally, I hear the advice to skip the upper few megabytes if you have problems enabling all your memory, or the machine locks up. E.g., enable only 78 out of your 80 MB. This is apparently the case for some SCSI controllers that use the very upper chunk of the main memory. Take it for what it’s worth.

Occasionally on some systems, Linux recognizes only 16 MB of memory. This is usually linked to the setting "memory hole at 15-16 MB" enabled in the BIOS setup (the solution is to disable this BIOS setting). It is probably a good idea to disable all "advanced" features in your BIOS setup anyway (for example, the BIOS virus detection seems to be a common source of problems).

Mixture of memory chips with different timings can also lead to memory recognition problems or to system crashes (the solution is to replace the memory chips so that the timing of all memory chips is the same).

4.1.2 LILO displays only LI (or LIL) and hangs

I quote from my good handbook "Red Hat Linux Unleashed" by Kamran Hussain, Timothy Parker, et al., published by SAMS Publishing:

"When LILO loads itself, it displays the word LILO. Each letter is printed before or after performing some specific action. If LILO fails at some point, the letters printed so far can be used to identify the problem. [...]

LI [...] This is caused either by geometry mismatch or by moving `/etc/lilo/boot.b` without running the map installer.

LIL [...] This is typically caused by media failure or geometry mismatch."

The geometry means the number of sectors/heads/cylinders used in the hard drive configuration of your BIOS. Hope this helps!

It is a very good idea to have a handbook for Linux or at least a general UNIX handbook. Handbooks for Windows are useless, handbooks for Linux are great! "Red Hat Linux Unleashed" is a very good handbook but I am sure there are many other equally good ones.

With a LILO error like above, you can boot your machine using a Linux or DOS boot floppy. There seems to be several general possibilities to correct such a LILO error, depending on what is wrong:

1. If LILO simply got corrupted, you can remove and re-install it. You can remove LILO by running under Linux:

```
lilo -u /dev/hda
```

or, under DOS:

```
FDISK/MBR
```

which rewrites the hard drive master boot record (MBR), in which LILO resides, and replaces it with "clean" DOS stuff. You will lose access to Linux, but you can boot it from the floppy and re-install LILO on top of the DOS MBR. To re-install LILO simply re-run the command `lilo` (as root).

2. Specify the option

```
linear
```

in your `/etc/lilo.conf` file. This is particularly useful for large drives (>8 GB). See `man lilo.conf` for details. The option "linear" is safe--it should not affect a properly working system, so you can specify this option is any case.

3. Look into your BIOS setup and figure out how your hard drive geometry does not match your hard drive. From under Linux, you can display the hard drive geometry using (for example, for the first IDE hard drive):

```
hdparm -g  /dev/hda
```

4. Put LILO on another partition (different hard drive) and make this partition bootable from fdisk (if your system supports booting from another drive).

5. Maybe you prefer to use "loadlin" instead of "lilo". From under DOS, locate your CDROM and see the program `/dosutils/loadlin.exe` . It boots Linux from DOS.

4.1.3 How can I change the operating system that LILO boots on default?

There are typically two sections in the file `/etc/lilo.conf`. One starts with "image=" and it is relevant to Linux, the other starts with "other=" and this is the dos/win section. The first to appear in `/etc/lilo.conf` is booted by default. So to change the system that boots on default, swap the sections. Don't forget to re-run the command

```
lilo
```

after any changes to `/etc/lilo/conf` .

4.1.4 The LILO prompt stays too short (or too long) on the screen during the bootup

Add or adjust the line

```
delay=100
```

right before the first "image=" or "append=" statement in your `/etc/lilo.conf` file. The number is the time of delay in tenths of a second (0.1 s), so in the example above the delay will be 10 seconds. Don't forget to re-run `lilo` after making the changes to the `/etc/lilo.conf` file, or the delay will not change.

4.1.5 Uninstalling Linux

If you really wanted to "uninstall" Linux, you can run this command (from under DOS or MS Windows):

```
FDISK/MBR
```

which will get rid of LILO--it overwrites the master boot record (MBR) of your first hard drive, where LILO resides. After this you can remove the Linux partitions using the DOS "FDISK" utility to re-claim the hard drive space.

4.2 Accessing my drives

4.2.1 Where are my drives?

Linux shows all the directories in one directory tree, irrespectively of what drives/hardware they are found on. Generally, this is a much better solution than the traditional DOS/Windows model--it completely abstracts the file system from the underlying hardware. You will appreciate this if you ever have to re-arrange or expand your hardware or add network resources. But for the users who are accustomed to the DOS way of dealing with drives, it adds some extra complexity.

In short, don't search for drive letters. There are none under Linux; the content of your disks appears as subdirectories on your single Linux filesystem (directory tree). On default, the content of

removable media does not appear automatically in these subdirectories--you have to "mount" your drives. See the next answers for details. You should also unmount a drive before ejecting the media.

You can access (read and write) a variety of drives and file systems from under Linux. This includes native Linux partitions, DOS and MS Windows partitions (on hard drives or floppies), ZIP and Jazz drives, and CDROM disks. Many less common file system types are also supported. This means that you can download your Linux software using Netscape for Windows, save the downloaded file on you MS Windows hard drive partition, and then boot Linux and copy the downloaded software from the Windows partition on you harddrive to the Linux partition, and finally install the software under Linux.

4.2.2 How can I access my CDROM?

Mount it. The mounting adds all the directories and files from your CD to your Linux directory tree so you can easily access them without the drive letter.

As root, you can mount the CDROM with a command like this:

```
mount -t auto /dev/cdrom /mnt/cdrom
```

If this works, the content of your CD appears in the directory /mnt/cdrom

Chances are this command will not work for you right away--you may have to customize it. Here is how it works.

The command tells the operating system to mount a filesystem autodetecting the filesystem type ("-t auto"). The device is /dev/cdrom. The mountpoint (the directory where to which "mounting" takes place) is /mnt/cdrom. This directory must exist and be empty. If it does not exist, create it with:

```
mkdir /mnt/cdrom
```

If the mounting command fails, make sure that the device /dev/cdrom exists. If it doesn’t, where is your CDROM? Chances are it is something like /dev/hdb if you have an IDE CDROM. Try /dev/hdb instead of /dev/cdrom in the mount example above. If this fails, you can try /dev/hdc or /dev/hdd, if your cd is an IDE CDROM. If none of them is your CDROM, maybe you don’t have IDE but a SCSI CDROM? Then try /dev/sda1, dev/sda2, etc. ["hda" is the the primary IDE master drive, "hdb" is the primary IDE slave drive, "hdc" is the secondary IDE master (if you have two IDE interfaces on your computer), hdd is the secondary IDE slave, "sda" is the first SCSI interface and the number is the SCSI device id number.]

It is a good idea to have a device `/dev/cdrom` anyway because some programs assume that it exists. If it does not exist on your system, you may create it as a symbolic link using, for example:

```
ln -s /dev/hdb /dev/cdrom
```

if your cdrom is the /dev/hdb drive.

If you cannot mount because "the device is already mounted or directory busy", perhaps the mountpoint `/mnt/cdrom` is your current directory. You have to change the directory to somewhere else in order to be able to mount to it; for example change the current directory to the root directory by issuing this command:

```
cd /
```

To unmount a mounted CD, exit the directory /mnt/cdrom and type as a root:

```
umount /mnt/cdrom
```

Your CDROM may refuse to eject the media if it is not unmounted. Also, you may have problems mounting the next CD if the previous one was not unmounted. If you cannot unmount because "the device is busy", perhaps /mnt/cdrom (or any subdirectory underneath) is your current directory? You need to change your current directory to somewhere else out of the mountpoint in order to unmount the device.

4.2.3 How to mount a floppy, zip drive, DOS/Windows partition, or a network drive?

Very much the same as CDROM--see the previous answer if you did not read it.

Floppy. I can mount my floppy (as root) with:

```
mount -t auto /dev/fd0 /mnt/floppy
```

Again, make sure that the directory /mnt/floppy exists and is empty. Also, `/mnt/floppy/` cannot be your current directory.

After a successful mount, the files from the floppy appear in the directory`/mnt/floppy/` . All the users will be able to read the files, but only root will be able to modify/delete the files. Please read further if you wanted the users to be able to write to the floppy.

To unmount a floppy (you *must* do this before ejecting the disk!) use:

```
umount /mnt/floppy
```

If you cannot unmount because "the device is busy", perhaps the `/mnt/floppy/` directory is your current directory. Exit it by typing (for example):

```
cd
```

which will change your current directory to your home directory.

Zipdrive. I mount the parallel port external zipdrive (scsi emulation) with:

```
mount -t vfat /dev/sda4 /mnt/zipdrive
```

The "-t vfat" is used here because zip disks come preformatted in the vfat filesystem, which is the filesystem of MS Windows with the long filename support. You won't be able to eject the disk without unmounting it.

All zipdrives (internal SCSI and IDE, external SCSI and parallel port) but the USB are supported under Linux (April 1999).

DOS/Windows partition. I use a dual boot system with both Linux and MS Windows on the same computer. I can access files on the DOS/Windows partition after mounting it with the following command:

```
mount -t vfat /dev/hda1 /mnt/dosdrive
```

Again, you may have to customize this command depending on what partition your DOS filesystem is. The "hda1" means the first IDE hard drive (hd a), first partition (1); "hda2" is the first IDE hard drive, second partition; "hda3"--the first IDE hard drive, third partition; "hdb1"--second IDE hard drive, first partition (or just "hdb" if it is the CDROM installed as a slave on your first IDE interface). "hdc" is the third IDE drive, hdd is the fourth IDE drive. SCSI drives have analogous names but start with letters "sd", followed by the letter indicating the SCSI interface, followed by the number indicating the SCSI device id . For example sda4 means "first SCSI interface, id number 4".

To mount so that all the users can read and write, you may want to try:

```
mount -t vfat -o user,rw,exec,umask=000 /dev/hda1 /mnt/dosdrive
```

This uses options (-o user,rw,exec,umask=000) to give absolutely everybody all the permission to all files on your DOS /dev/hda1 partition (you should ask yourself if this is really safe on your system). If users still can't write to the DOS partitions, perhaps the permissions on your mountpoint need to be set. This command (executed by root) will set up the permissions on the mountpoint `/mnt/dosdrive` so that all users will be given rights to read, write and execute:

```
chmod a=rwx /mnt/dosdrive
```

Network File System (NFS). This is great for direct access to files that reside on another Linux computer. For mounting of a remote filesystem as NFS, first check if the NFS service is enabled (use the program `setup`). NFS also requires permission from the other computer. To configure the permissions on the server machine, run as root:

```
netconf
```

and adjust the setting under "Exported File Systems" menu.

If you prefer to do it manually, the permissions are set in the file `/etc/exports`. My `/etc/exports` looks like this:

```
/usr hacker(ro) mars(ro)
/home hacker(rw) mars(rw)
/mnt hacker(rw) mars(rw)
```

This gives the machines called hacker and mars the permission to mount the directories `/usr/` (read-only access), `/home` and `/mnt` (read-write).

If you set up your NFS properly, you should now be able to mount a network directory using a command like this:

```
mount -t nfs mars:/home /mnt/mars_home
```

This mounts the contents of the directory `/home/` on a machine called "mars" into the directory `/mnt/mars_home/` (which must exist and be empty).

Many operating systems know NFS, but MS Windows doesn't. Therefore MS Windows remote shares have to be dealt with differently. See the next answer for details.

4.2.4 How to mount a remote MS Windows filesystem through Samba?

A remote MS Windows filesystem can be mounted onto a Linux filesystem through the Samba protocol (Samba must be installed, go here [p 86] if it isn't). Type a command like this (as root):

```
smbmount //mars/windows /mnt/mars_windows -c marie
```

This mounts the MS Windows resource called windows from the MS Windows machine called mars. The mountpoint on the client computer is `/mnt/mars_windows/`. The option "-c" specifies that the samba server is a machine called marie (this should not be necessary, but it is on my system).

For the above to work, the permission must be given on the MS Windows machine for sharing the directory or drive as a resource. To do this, on the MS Windows machine, enable the filesharing using the "control panel-network", then launch the "Windows Explorer", click the right mouse button on the drive or directory to share, click on properties, switch to the page "sharing", give yourself the permission and give the resource a name.

To unmount an MS Windows directory use:

```
smbumount /mnt/mars_windows
```

If you have problems, see:

```
man smbmout
```

4.2.5 Any quick way to access a file on a DOS/Windows floppy?

Use "mtools", no mounting required. For example, I can use the mdir command to quickly inspect the content of the root directory on my DOS floppy:

```
mdir a:\
```

I can also use mcopy to copy the file "autoexec.bat" from the root directory on the floppy to my current directory on Linux:

```
mcopy a:\autoexec.bat .
```

You have to be root to be able to write to a floppy.

Type "mtools" to see the supported commands in the rich mtools set, which parallel the most popular DOS commands (for example: `mformat`, `mtype`, `mren`, `mmove`, `mdel`, `mrd`, `mattib`, ...), and use manual pages if you have problems using them. For example:

```
man mtype
```

will show me how to display a content of a text file on a DOS partition.

To access DOS drives other than a: or b:, you have to configure mtools so as to indicate which devices are associated with other DOS "drive letters". This is quite easy--you just edit and modify the file `/etc/mtools.conf`. I typically use pico to do it (as root):

```
pico /etc/mtools.conf
```

For example, my `/etc/mtools` contains a line like this:

```
drive c: file="/dev/hda1"
```

which instructs the mtools that the partition "/dev/hda1" will be called "c:" The setup of `/etc/mtools.conf` requires just uncommenting (removing the "#" at the beginning of the line) and adjusting the appropriate entry.

4.2.6 Mounting works when I am root. Can a normal user mount?

You have to edit the file `/etc/fstab` as root to give the normal users the permission to mount a particular drive. For example I can use the `pico` text editor to do this:

```
pico -w /etc/fstab
```

The option "-w" turns off the long line wrap.

Here is the content of my /etc/fstab:

```
/dev/hda2 /ext2 defaults 1 1
/dev/hdc3 /home ext2 defaults 1 2
/dev/hdc2 /usr ext2 defaults 1 2
/dev/hdc4 swap swap defaults 0 0
/dev/fd0 /mnt/floppy auto noauto,users,rw 0 0
/dev/cdrom /mnt/cdrom auto noauto,user,ro 0 0
/dev/sda4 /mnt/zipdrive vfat noauto,user,rw,exec 0 0
```

```
/dev/hda1 /mnt/dosdrive vfat noauto,user,rw 0 0
none /proc proc defaults 0 0
hacker:/mnt/cdrom /mnt/hacker_cdrom nfs noauto,user,ro 0 0
hacker:/mnt/floppy /mnt/hacker_floppy nfs noauto,user,rw 0 0
hacker:/home /mnt/hacker_home nfs noauto,user,rw 0 0
hacker:/usr /mnt/hacker_usr nfs noauto,user,rw 0 0
```

Each line contains six space-delimited fields (this means that each line has six entries separated by white space). The first field is the name of the device. The second field is the mountpoint (an existing directory on your Linux system to which the resource will be mounted). The third is filesystem type. For removable media that may contain filesystems of several types, I use the option "auto" to let Linux probe which filesystem is currently present there. (The order in which they are probed is determined by the content of the file `/etc/filesystems` . You may want to make sure that it specifies "vfat" before "msdos" or the long DOS filename may be cut short.) The fourth field contains options: "auto" = mount the filesystem on the system startup; "rw" = read and write allowed; "ro" = read only, "user" = users have the permission to mount this filesystem (one can also use "users" to allow a user to mount and another user to unmount--otherwise only the user that mounted the filesystem can unmount it), "exec" execution of programs is permitted from this filesystem. The number in the field 5 specifies if the filesystem is to be backed up during a system backup, the number in the field 6 determines if to check up the filesystem integrity during bootup. The hacker stuff in my `/etc/fstab` are filesystems on another computer (called "hacker") on my home network and it serves here as an example of how to mount network resources. Check `man fstab` for more info.

For example, if regular (non-root) users have the permission to mount the cdrom (the "user" option is specified), they can mount it using a command like this:

```
mount /mnt/cdrom
```

The command which the root uses for mounting (see here [p 63]) will not work for a regular user because the regular user is restricted by the options in `/etc/fstab` and therefore s/he cannot specify simultenously both the device and the mountpoint.

For a regular user to be able to write to a disk or execute a program on it, s/he must also be given the appropriate permission on the "mountpoint" directory. For example, this will give all the users all the permissions (read, write, execute) on the directory `/mnt/floppy` :

```
chmod a+rwx /mnt/floppy
```

Now (also the "rw" option is specified for the floppy in the `/etc/fstab`) the user will be able to write to a floppy. If the "exec" option was enabled in the `/etc/fstab`, the user would also be able to execute programs from the floppy.

Please note that the DOS vfat file system doesn’t know about the file permissions the way Linux does. Linux manages this during mounting by giving the default file permissions on the mounted filesystem: the user who mounted the filesystem will be the owner of all files and will be given the right to write to the filesystem (if "rw" was specified in fstab) but other users can only read. If you wanted to change this behaviour, you could use the "umask=" option so that the appropriate line in

your `/etc/fstab` may look like this example:

```
/dev/sda4 /mnt/zipdrive vfat noauto,users,rw,exec,umask=000 0 0
```

This gives absolutely everybody all the permissions on your zipdrive (mounting, unmounting, read, write, execute).

To summarize, the file `/etc/fstab` is the place to keep your defaults on how to mount filesystems and what kind of access is allowed for users. You really want to customize it to simplify mounting on your system. Linux default mounting scheme is restrictive so as to be secure, you may want to remove some restrictions when setting up Linux at home.

4.2.7 Mounting command is too long, how can I simplify it with an alias?

An alias is an abbreviation of a more complex or often used operating system command. For creating aliases, I edit, as root, the file `/etc/bashrc` . This way the aliases are available for all the users on the system. (For creating user-specific aliases, I edit the file .bashrc in the user home directory.) The relevant part of my `/etc/bashrc` looks like this:

```
alias cdrom="mount -v /mnt/cdrom"
alias ucdrom="umount -v /mnt/cdrom"
alias dosdrive="mount -v /mnt/dosdrive"
alias udosdrive="umount -v /mnt/dosdrive"
alias zipdrive="mount -v /mnt/zipdrive"
alias uzipdrive="umount -v /mnt/zipdrive"
alias floppy="mount -v /mnt/floppy"
alias ufloppy="umount -v /mnt/floppy"
```

The option "-v" stands for "verbose", i.e., it tells Linux to talk to me a lot during mounting. For the aliases to take effect, the user has to re-login. Now the user can mount the floppy using this simple command:

```
floppy
```

and s/he can unmount it using

```
ufloppy
```

4.2.8 Can I automount?

You can. Automount will automatically mount a filesystem as you access it and unmount when you stop using it.

To set up automount, I first run the programs `setup` (as root). Go to the option "netsysv" (RH5.2) or "System Services" (RH6.0) and make sure that automount service ("autofs") is enabled.

Then, I configure automount by editing the files `/etc/auto.master` and `/etc/auto.misc`, e.g. (as root):

```
pico /etc/auto.master
```

My /etc/auto.master looks like that:

```
/misc /etc/auto.misc --timeout 1
```

This says that my automount devices will be mounted in the directory`/misc` (which must exist and be empty). My automount drives will automatically unmount one second after I stop using them (for example, after I exit the directory). This is a short time--you may choose a longer one. The detailed config file is `/etc/auto.misc` . Here is mine:

```
kernel -ro,soft,intr ftp.kernel.org:/pub/linux
cdrom -fstype=auto,ro :/dev/cdrom
floppy -fstype=auto,rw :/dev/fd0
zipdrive -fstype=vfat,rw :/dev/sda4
dosdrive -fstype=vfat,ro :/dev/hda1
hacker_cdrom -fstype=nfs,ro hacker:/mnt/cdrom
hacker_floppy -fstype=nfs,rw hacker:/mnt/floppy
hacker_usr -fstype=nfs,ro hacker:/usr
```

Each line consists of 3 space delimited fields. The first field is the "key" which will be the name of the subdirectory (under /misc) to which the device will be mounted. This directory must NOT exist. It will not be visible when I use the command `ls` , but I can "cd" to it and my device will then mount. Don’t ask me why it is so, and how to use this automount in GUI. I don’t know. The hacker stuff in my auto.misc is the cdrom and floppy from another computer in my home network.

I automount to the directory `/misc` (not `/mnt`) so that I can also mount filesystems manually, without using automount, to the directory /mnt.

4.2.9 How do I get my parallel-port (external) Zip drive recognized?

RedHat 5.2 During the initial RH5.2 installation, answer "yes" to the question "do you have scsi devices" Then pick up the parallel port zip drive (ppa) from the list. During the disk partitioning later in the installation process, either have a disk in your zip drive or tell the install process to "skip" when it attempts to access your zip drive.

If you didn’t setup your Zip drive during the initial RedHat installation, you may also add the Zip support later in a way similar to adding the network card or any other module to the kernel:

As root, start an X-terminal, and run

```
/usr/bin/kernelcfg
```

to insert the ppa module into the kernel. (The "ppa" stands for, I guess, the "parallel port first interface ’a’ ".)

After everything is done, inspect `/proc/modules` to see if the ppa module is loaded (or run as root `lsmod` to list the loaded modules). The information from `kernelcfg` goes to the file `/etc/conf.modules` so if you have difficulty removing modules (e.g. inserted by mistake)--as I did--just edit and adjust this file manually.

Now your zipdrive should be recognized. Put a zipdisk into it and try mounting using the previously described command (as root):

```
mount -t vfat /dev/sda4 /mnt/zipdrive
```

Using the Linux kernel ver.2.0.36 and below, you will not be able to use a printer which is connected through the parallel port zip drive (this works with the kernel 2.2.x). Use the command

```
uname -a
```

to see which version of kernel you are running.

RedHat 6.0 and 6.1 The zipdrive (zip100 drive) installation did not work during my upgrade to RedHat 6.0 (the installation program said that it couldn’t find the zipdrive). So, after the installation was completed, I issued the following commands to insert the modules for parallel port zip drive into the kernel (as root):

```
/sbin/insmod parport
/sbin/insmod ppa
```

To have these two lines executed automatically after each bootup, I added them at the end of the file `/etc/rc.d/rc.local` (this file is something like autoexec.bat on DOS).

If this still does not work for you, you may also want to edit the file `/etc/conf.modules`. Mine contains such a line:

```
alias parport_lowlevel parport_pc
```

and there is no line mentioning the "ppa" module.

For the newer Zip250 drive, I have the following two lines executed from my `/etc/rc.d/rc.local` file:

```
/sbin/insmod parport
/sbin/insmod imm
```

4.2.10 Can I set 32-bit hard drive I/O?

I know nothing about harddrives, but saw a review of RedHat 6.0 by Edward Choh (http://hardwarezone.community.com.sg/main.htm), in which he gave some interesting tips, including setting up the 32 bit input/output (I/O) and direct memory access (DMA). I tried it, and it worked fine for me. Be warned that it can possibly harm the content of your harddrive, so do not do it if you are a real PC newbie, don't feel geeky today, or have a weak heart--I can't guarantee it will work for you.

The turning on of the 32-bit I/O and DMA has to be done by root, and I did it in a single-user mode (to minimize the damage to the file system if something went wrong and I had to reboot). I definitely would not do it on a system currently running many programs or X-windows, and would have a current backup of any precious data.

To boot your computer in a single-user mode, I type this at the lilo prompt (during bootup):

```
linux single
```

Say I would like enable the 32-bit I/O on my first IDE harddrive, which is "hda". First, I time the harddrive current performance, and note the score:

```
hdparm -t /dev/hda
```

Now, I display my current I/O and DMA settings:

```
hdparm -c /dev/hda
```

[my system showed 0, meaning that the 32-bit I/O is turned off and the default 16-bit access is used]

```
hdparm -d /dev/hda
```

[my system showed 0 again, meaning that the harddrive DMA access is turned off].

Now, I turn on the 32 bit IO and DMA:

```
hdparm -c 1 /dev/hda
hdparm -d 1 /dev/hda
```

Now, I can time the performance of the harddrive again to compare the score with the original one:

```
hdparm -t /dev/hda
```

If everything worked ok, and the performance has improved, I can "commit" the new settings, so they can survive a soft reset:

```
hdparm -k 1 /dev/hda
```

To have the new settings in effect every time you reboot the machine, you may add a line at the end of the file `/etc/rc.d/rc.local` (this file is something like `AUTOEXEC.BAT` in DOS):

```
hdparm -c 1 -d 1 -k 1 /dev/hda
```

If something did not work as expected, or the performance did not really improve, I can reboot at any time and the old settings will be in effect as long as I did not perform the last operation.

I performed this tune-up on 4 hard drives on our home network. It was a success on 3 newer hardrives: the performance improved by 30-300% and at least one computer "feels" faster than before. One harddrive (which is always flaky) hanged the computer hard during the performance test and I had to reset the machine (no damage done).

4.3 Working with X-windows

4.3.1 How do I install kde on RedHat 5.2?

RedHat 5.2 does not install kde by default because at the time of the RH5.2 release there was a problem with the license for a library that the kde uses. Otherwise, kde is a very good GUI, the library license problem is now solved, and the latest RedHat (6.0) includes kde (and the more ornamental GNOME). Still, the kde binaries are on your RedHat 5.2 CDs, you just have to install them yourself. (If you don’t have the CDs, you can download the binaries from the kde site on the Internet.)

First check if your X-server works by executing:

```
xinit
```

[The X-server is a bare-bone X-windows system, without a "Windows Manager." You can execute your X-windows programs from here by typing the program name (with leading ./ or full path) in the X-terminal window, but you will not be able to move or resize the windows, add icons, etc.]

You can exit your X-server by typing exit in the X-terminal window, or pressing <Ctrl><Alt><Bkspace> to kill the X-windows server. (The last solution is perfect should your windows ever hang-don’t reboot in such a case.) If your X-server does not work, see the next answer.

Now, login as root. Mount the RH5.2 CD:

```
mount /mnt/cdrom
```

Go to the proper directory:

```
cd /mnt/cdrom/kde/distribution/RedHat/i386/binary
```

It is useful to use the command line autocompletion (press Tab) when typing long paths or filenames.

Read the README file:

```
cat README-2rh51-rpms | more
```

Use the `rpm` "RedHat Package Manager" to install the necessary packages

```
rpm -iv packagename
```

The packages have filenames ending with .rpm. First install the qt libraries, then kde support, then kde libs, then kde base. If you choose the wrong order, the dependency check will fail and the package will not install (rpm will issue a message). This is not serious, just re-install the required package first, and then try the next package again. After you are done with the base, you may want to install all other packages for kde (utill, admin, network, games, graphics, multimedia)--they are not big, so you may consider installing them all. Finally, just to make sure that you installed everything type:

```
rpm -ivh  *.rpm
```

The options "vh" print some extra info. You will get some messages like "the package is already installed" If there is more than a screenful of them, you can scroll back using <Shift><PgUp>. If you really don't like the command-line-based rpm package manager, you may install the same packages using a GUI front to rpm called glint (available only in RH5.2). Just type "glint" in the X-windows terminal.

Now tell your system that kde is to be your default X-windows manager. In user home directory, create an .Xclients file:

```
pico .Xclients
```

which contains just one line:

```
/opt/kde/bin/startkde
```

Type in the line and save the file. (Adjust the line as requierd so the location of the `startkde` file is correct.) Now, make the file executable to all users:

```
chmod a+x .Xclients
```

Check if the permissions were changed:

```
ls -l .Xclients
```

[Files with a dot at the beginning are not displayed by a regular ls command, there are something like hidden files under DOS. You must use its name or `ls -a` .]

If you created the file as root not the user, change the owner and the group of the file to the proper user:

```
chown user_name .Xclients
chgrp user_name .Xclients
```

That's it. Now typing `startx` should start your X-server with the kde as the windows manager.

4.3.2 How do I setup video card, monitor and mouse for the X-server?

This should be set-up during your RH5.2 Linux installation unless you skipped the step.
To set it up now, you may try, as root, one of these configurators:

```
Xconfigurator
XF86Setup
xconf
```

The simplest way is to run the command `setup` (as root) and access Xconfigurator from there.

To setup X-windows under Linux, you may need to know your hardware. You may want to dust your monitor manual to see what max synchronization frequencies (vertical and horizontal) your monitor supports. The message when the computer boots may give you a clue about what type of video card you have and with how much memory. Read the label underneath your mouse to find out about the mouse. Next time you buy a mouse, get a 3-button "Linux-ready" Logitech or similar (Linux makes a good use of all three mouse buttons).

When testing the X-server, if the screen goes blank, displays funny lines, or otherwise obviously does not function as designed, kill it fast with <Ctrl><Alt><BkSpace> and re-check your monitor sync frequencies. Running wrong frequencies can be bad for your monitor.

If you really have problems, set up a basic vga (640x480) server first. You can fine-tune it later, after you get some understanding of how things work on your system.

4.3.3 How to X-window remotely?

- Start X-server on the local machine, e.g.

```
xinit
```

- From the x-terminal give the remote machine the permission to display on your local screen:

```
xhost name_of_the_remote_server
```

In the really secure environment of my house, I could even give all servers the permission to display on my screen using:

```
xhost +
```

- Telnet the remote server.

- Start an X-program on the remote server directing the display on your local screen, for example, you may start a window manager:

```
startkde -display local_machine_name:0.0 &
```

The symbol "&" puts the command in the background, so that your telnet window is still available to you.
The 0.0 means "display zero, screen 0", which is your first screen on the first display and makes sense since you can have many concurrent sessions of X running on your computer with Linux.

You don’t have to specify the "-display" option if your environment variable DISPLAY specifies the correct location on your current terminal, which is the case on my systems by default, but not on everybody else’s as I am told. You can check your DISPLAY setting using:

```
echo $DISPLAY
```

- After I finish my remote X session, I restore the access control to my X-server using:

```
xhost -name_of_the_remote_server
```

or

```
xhost -
```

Example. This sequence of commands will run Netscape on the remote machine called marie, directing the display to the X-server with X-windows manager which runs on the local machine hacker:

```
startx
xhost marie
telnet marie
[login]
netscape -display hacker:0.0 &
[do my stuff]
[logout]
xhost -marie
```

In principle, you can run a program on any computer on the network, and display the output on any other (not necessarily the one you are sitting at).

I use remote X-windowing a lot to run fat programs (kde, Word Perfect 8, and Netscape) on a slim machine (486-33, 8 MB mem) which would not be able to run those by itself. It is also a convenient and fast way to work with files on a remote system for which the nfs mount is not set up.

4.3.4 Can I have multiple sessions of X running at the same time?

Yes, you can. When you issue the first `startx` command on your system, it opens the first X-session on the default display 0. The second X-session must be opened on a different display. For example, this will open a second X-session on the display 1:

```
startx -- :1
```

You can have up to 6 concurrent X sessions. Use <Ctrl><Alt><F7> to switch to display 0, <Ctrl><Alt><F8> to second screen, etc. up to <Ctrl><Alt><F12> Try `man startx` if you need more info.

In a similar fashion, you can open another bare X-server session without a window manager. This will open one on display 2:

```
xinit --  :2
```

On this bare-bone X-display I can run a different windows manager (so as to have two different ones running at the same time) by typing in the X-terminal window one of these:

`fvwm2`	(to run fvwm2)
`startkde`	(to run kde)
`afterstep`	(to run afterstep)
`fvwm`	(to run fvwm)
`AnotherLevel`	(to run AnotherLevel)
`gnome-session`	(to run Gnome)

Of course the alternative windows manager will run only if it is installed on your system. The above windows managers are available on your RH5.2 CD for you to decide if you want them (with the exception of Gnome, which is not present on RH5.2 but is the default on RedHat 6.0 and 6.1). I use almost exclusively KDE, although the other managers may be smaller and faster. Gnome is a famous X-windows project which is said to be more advanced and is prettier than KDE, but it is still quite buggy, so perhaps not recommended unless you don't mind occasional trouble. RH6.0 and 6.1 contain both major X-windows systems, Gnome and KDE.

4.3.5 Can I have a GUI login prompt?

To start up your X-server automatically on the system start-up and display a graphical login prompt, you have to edit just one character in the file `/etc/inittab` . This file specifies something like:

```
id:3:initdefault:
```

Change it to

```
id:5:initdefault:
```

This sets up the default runlevel to 5, which is X-Windows. The meaning of the different runlevels is explained in the same `/etc/inittab` file :

```
0 - halt (Do NOT set initdefault to this)
1 - Single user mode
2 - Multiuser, without NFS (The same as 3, if you do not have
networking)
3 - Full multiuser mode
4 - unused
5 - X11
6 - reboot (Do NOT set initdefault to this)
```

You can change the runlevel from the command line with a command like this (as root):

```
init 6
```

which will reboot your computer.

4.4 Printer and soundcard

4.4.1 How to setup my soundcard?

Try to run (as root)

```
setup
```

The third button from the top ("sndconfig") sets up the sound card. Unless you have a very fancy sound card, this will work for you. At the end of the setup, Linus says how he pronounces "Linux". ("sndconfig" can be also run directly from the command line--type `sndconfig`.)

You may want to try your soundcard and cdrom using a command line cdplayer. Put a music CD to your CDROM and type:

```
cdplay
```

If this does not work, maybe you don't have `/dev/cdrom`? Check if you can mount a data CD as a root (look here [p 62]) and create the device `/dev/cdrom` by linking it to the appropriate drive (most likely `/dev/hdb`), for example:

```
ln -s /dev/hdb /dev/cdrom
```

If `cdplay` works for root, but does not work for a regular user, maybe you need to give (as root) everybody the permissions to read and write to the the file `/dev/cdrom` :

```
chmod 666 /dev/cdrom
```

(The directory `/dev` is where all your devices appear as files.)

Now, that you got hooked on `cdplay`, you can play third song, try:

```
cdplay play 3
```

You can also use the command `cdp` for rudimentary command line interface to cdplay, but perhaps you preffer the interfaces available from under X-windows (e.g., from KDE "K" menu, choose: Multimedia-"CD Player").

To stop the music either press the button on you CDROM or issue one of these commands:

```
eject
cdplay stop
```

4.4.2 How do I setup my printer?

As root, start an X-windows session by typing

```
startx
```

Read here [p 74] if your X-windows is not set up properly yet.

Start an X-terminal (the "two-computer" button on your K-bar) and type in it:

```
printtool
```

This program does a complete printer setup, you just have to fill up the information about your type of printer and where it is hooked up.

Specifying the proper printer port is the most important part. If you don't know which one is yours try:
on RedHat 5.2: lp1 (this is the first parallel port on RH5.2) or lp2 (this is the second parallel port on RH5.2) or lp3 (this is the third parallel port on RH5.2); **on RedHat 6.0:** lp0 (this is the first parallel port on RH6.0) or lp1 (this is the second parallel port on RH6.0) or lp2 (this is the third parallel port on RH6.0). After upgrading from RH5.2 to 6.0, the printing stopped working because of the names changes of the parallel ports. I had to re-run the printool and adjust the port.

Try printing an ASCII test-page straight to the port. Only when this works set up the bells and whistles.

If you are setting up a remote printer, make sure that your machine has the permission to use the remote printer. The permissions are set in the file `/etc/hosts.lpd` (more secure) or `/etc/hosts.equiv` (less secure). These files simply list the names of the machines, one machine per line. Mine looks like this:

```
hacker
mars
```

The file `/etc/hosts.lpd` did not exist on my system, so I created it.

For quick information about the printers on your machine, you may want to view the file `/etc/printcap` :

```
cd /etc/
cat printcap
```

Here is the meaning of some codes that I see in my `/etc/printcap`:

:	Field separator (separates the entries in the file).
\	(at the end of line) Continuation on the next line.
lp	Name of the printer. "lp" is the name of the default printer on your machine. Subsequent printer are often, by default, given the the names lp0 or lp1, ... (or whatever you like) but this should not be confused with the name of the devices (ports) to which they are connected.
sd=/var/spool/lpd/lp	My spool directory (sd).
mx#0	Maximum size of print jobs (mx) in blocks. "0" means no limit.
sh	I want headers to be suppressed (sh). Header is the page with your name that prints before your printing job (waste of paper if you print at home).
rm=mars	Name of the remote machine (rm), which on my system is called "mars (my printer is connected to a different computer).
rp=lp or lp=/dev/lp0	Name of the remote printer (rp), which is the name of the printer on the remote machine ("lp" on my system) or the name of the device on the local machine. "/dev/lp0" is the first parallel port on RH6.0 (it used to be /dev/lp1 on RH5.2).
if=/var/spool/lpd/lp/filter	Input filter (if). Your printing job will be formatted by this "filter" before it is sent to the printer.
sf	Suppress the form feed (sf) that is normally sent when printing is completed (use it if your printer keeps printing an empty page at the end of each jobs).

The printer is controlled using the command `lpc` (as root). Type "?" to see the options. This program is notorious for its peculiarities, so don't get discouraged easily. The printer queue can be viewed with `lpq` and cleaned with `lprm`,both of which work for a user (not only root). You can print from the command line using the command `lpr`.

4.4.3 Word Perfect 8 does not have a driver for my printer

If you installed your printer in KDE using the printtool and it had a driver which works alright, set up Word Perfect to print using the "passthru postscript" driver.

4.5 Setting up a network

4.5.1 Would it be worth it to set up my home network?

This is an excellent idea. It will let you use the machines that are in your closet now because they were not powerful enough to run standalone. The benefits include sharing hard drives, zipdrives, CDROMs, modem, printers, even soundcards, running programs remotely (the text or graphics mode), browsing the Internet on all computers at the same time over one phone connection. If you ever lose control of your machine, you may also be able to shut it down remotely and thus avoid possible problems--see the answer on shutting down for details.

4.5.2 How to set up my home network?

Hardware. Your hardware must be set up properly. Your network card should have been set up during the initial RedHat installation. If it wasn't, you may want to set up your network card now from X-windows. If you have more than one network card on your computer, you must also set up the second cards manually, since Linux will autodetect only 1 network card.

Start an X-terminal, execute "su" (to make yourself a root) and then run:

```
/usr/bin/kernelcfg
```

Under Linux, most drivers for network cards are implemented as modules. So setting up a card involves just inserting the proper module with its parameters. You don't need to recompile the kernel, nor even reboot your computer (unless you have an uncommon card). To figure out what module(s) and parameters you need, see the Linux Network Administrator Guide (file `/usr/doc/LDP/nag` on your system or check http://metalab.unc.edu/mdw/index.html#guide). This excellent guide is known as nag.

The parameters (IRQ and address) were a problem during my setup of two network cards on one computer--you must make sure that you don't have any hardware conflicts. A common source of problems is that the card wants to configure on IRQ 5, which is occupied by the SoundBlaster, or IRQ 3 which conflicts with the second serial port (COM2, cua1, ttyS1). Inspecting the files `/proc/interrupts`, `/proc/ioport` and reading bootup messages may help.

For example, my WD8013 card (same as SMC Elite and SMC Elite plus, according to nag) runs under IRQ 10 (set by a jumper on the card and I specified the IRQ in the kernel module setup) under the address 0x300 and uses the "WD" module. My "SMC EtherEZ" card (no jumper settings on the card) runs under IRQ 9, address 0x240 and uses "SMC EtherEZ" module. Please make sure you don't omit the leading "0x" in the address--it means "hexadecimal" and must be there else the number will be interpreted as decimal.

After the module is added from `kernelcfg` and the kernel is re-run (`kernelcfg` provides this, see the menus), inspect the file `/proc/modules` to see if the module is indeed loaded (or run `lsmod` as root). Information from `kernelcfg` goes to the file `/etc/conf.modules` so if you encounter difficulties (for example, I had difficulty removing modules inserted by mistake), just edit and adjust this file manually, e.g., using `pico`.

Network. After setting up the network cards and connecting the cables, set up the network by running (as root):

```
netconf
```

This program has help! `netconf` can also be run under GUI, but I did not really try it. If you need more understanding of how networking works, you may want to read the previously mentioned nag.

How and what to fill up in `netconf` depends on your network. You surely want to fill up "Basic Host Information" (enable the first ethernet interface, eth0, fill in the name, aliases and IP of your local computer) and the "Information About Other Hosts" (names, aliases and IPs of other computers on your local network). This information goes to the files: /etc/hosts and /etc/sysconfig/network, so you may want to inspect these files and adjust them manually.

When setting up the network, don't mess up with the "loopback driver" which has the IP 127.0.0.1. It is always there--it is the IP through which the machine talks to itself.

If you don't have an IP address (as will typically be the case for a home network with no permanent connection to the outside world), you may want to invent one. It does not matter very much what it is since when connecting to the outside world by your modem, you will be dynamically allocated an IP address (a machine can have many IP addresses at the same time). Your invented IP must be formally correct and the net mask must match the class of the network (class A, B or C). See the chapter on IP addresses in the already mentioned Network Administrator Guide (nag) (`/usr/doc/LPD/nag` on your system, or if you don't have it, download it now from http://metalab.unc.edu/mdw/index.html#guide.

Typically for a home network, you want a class C network (up to 254 machines, the smallest) which has the first three digits between 192 to 223. The last three digits identify the machine on your class C network and must be between 1 and 254 (don't use 0, it means "whole network" or 255 which is the "broadcast address"). The middle two sets of digits can be anything from 0 to 255. Thus 223.223.223.1 is the first machine on the class C network 223.223.223.0 (the last zero signifies the whole network), with the broadcast address 223.223.223.255. The network mask for a class C network is always 255.255.255.0 (unless you subdivide your network into smaller "subnet", which is not discussed here).

Typically, the first machine on a network is the one that is expected to have the connection to the outside world (since it was connected first, but there is no standard for that) . So, I filled up the GATEWAY to xxx.xxx.xxx.1 (my first machine) on all machines, except for the machine xxx.xxx.xxx.1, where I left this field blank. Actually, although I left the field blank, netconf inserted the gateway 0.0.0.0 into the /etc/sysconf file. This was a source of an annoying error message during the loading of the network card on the bootup. To get rid of the message, I edited /etc/sysconfig/network and set it to something like this:

```
GATWAYDEV=
GATEWAY=
```

[You don't want a gateway on your ethernet interface on this machine if it does not lead to the outside world. The ppp interface on this first machine will be set up as a default gateway once you connect through your modem, e.g. using kppp.]

The name of a computer is entirely arbitrary--the main user normally chooses a short word s/he likes. The domain name of the home network is also entirely arbitrary, unless you have a permanent connection in which case a domain name is registered to you. Try to invent something that does not exist yet--it could make your life easier once you have a permanent connection.

As for the DNS server (also called "named"), RedHat 5.2 and 6.0 comes with DNS preinstalled as "caching-only" so it is easy to configure as such. You may also choose not to use DNS at all--if DNS is looking up a server and can't find it, it can be a real show-stopper (the machine can appear to be hanged for up to a few minutes). To use DNS, the "named" service must be enabled-- check this by running `setup`. To set up the caching-only DNS, fill up the appropriate boxes in `netconf`. E.g., I filled "nameserver 1" to my first machine on all computers (I entered the loopback address 127.0.0.1 on the machine xxx.xxx.xxx.1, and the proper xxx.xxx.xxx.1 address on all other machines).

No routes to other networks and hosts were required in my network, since I don't have other local networks. So I left this field blank.

Other than setting up the hardware correctly and filling up the info under `netconf` on each computer, as described above, I did not have to do anything on the standard RedHat to get my network working.

Reboot all computers one by one in any order (this is not necessary, but won't hurt you) and watch the boot messages (if they scroll too fast off screen, use <Shift><PgUp> to scroll up, or use `dmesg` from the command line to view them latter). Did your cards configure correctly? Use the command `route` (as root) to see if the eth0 interface is running. Use the ping command to test the connections between individual machines. Try to telnet your local machine to see it the loopback works:

```
telnet name_of_the_machine_you_are_sitting_at
```

After a successful login, you can exit the telnet session by typing

```
exit
```

Finally, try to telnet a remote machine on your network:

```
telnet name_of_a_remote_machine
```

If this works on all machines, your eth0 network interface is set up.

After setting up your ppp and connecting to your Internet Service Provider (ISP), you will have another network interface (ppp0) and then will be able to telnet any machine in the world.

4.5.3 I have problems configuring my ppp dial out

GUI. If I were you, I would install kde and use kppp. (If you don't have kde installed, look here [p 72] for information on how to install it.) kppp is really easy to configure and run. To set up your ppp dial out, run kppp (it is under the "K" menu "Internet, or you can run it from the xterm by typing `kppp`), press the "setup" button, create an account, and fill out the information required: telephone number, authentication protocol, and your Internet Service Provider's (ISP) domain name and the Domain Name Server (DNS) number of your ISP. All this information should have been provided to you by your ISP.

Next, I created a device called /dev/modem, by symbolically linking it to the device /dev/ttyS1 (as root)

```
ln -s /dev/ttyS1 /dev/modem
```

ttyS1 should work if your modem is on the port that DOS calls COM2. Use ttyS0 for COM1 and ttyS2 for COM3. The modem will not dial at all if a wrong port is chosen.

Finally, I tried to run the kppp but got an error message complaining about a "lock". So, I made sure that the file `/etc/ppp/options` is empty by editing it (as root):

```
pico /etc/ppp/options
```

and deleting the word "lock", then saving the file.

After these simple steps, the kppp could connect to my ISP with no problems on RedHat 5.2.

RedHat 6.0 required one additional step: setting the "suid" ("substitute user id") so that "kppp" runs with the effective user id of root (because it needs to access hardware directly). Without it, kppp complains that "it was not properly set up" and "can't create lock file". This has to be done as root:

```
cd /usr/bin/
chmod a+s kppp
```

Troubleshooting. If your modem refuses to dial on the port that you are positive is chosen properly, maybe the modem is not set up properly (or maybe it is a "winmodem"? Then throw it away and buy a proper modem). In one instance, I had to run kppp setup, edit the "modem commands" and input ATZ1 as the initialization string (instead of the default ATZ for a standard Hayes-compatible modem).

If your telephone line requires pulse dialing (instead of the default touch-tone dialing), you may need to change the modem dialup command from "ATDT" (="ATantion Dial Touchtone") to "ATDP" (="ATantion Dial Pulse"). These commands work with any standard Hayes-compatible modem.

If your modem dials correctly and you are able to connect, but your authentication fails, perhaps your Internet Service Provider (ISP) uses a different authentication protocol. Call them and ask what authentication protocol they use. Or try pap, terminal-based or chap (in your kppp setup) until you

find one which works with your ISP.

In one instance, I had a problem with the reliability of establishing a connection (the error would pop up saying something like: timout for the pppd startup", and the connection would establish only once every few trials). The problem was solved by changing the "flow control" option (in the kppp "setup" under "device") from "CRTSCTS" to XON/XOFF. (Still CRTSCTS is the recommended flow control method in most cases.)

The performance (speed) and the reliability of the connection with my kppp was much improved after I upgraded from RH5.2 to RH6.0. So if you have problems with these, maybe it is time to upgrade.

If you keep having problems setting up ppp, you may want to try `minicom` to see if you can get your modem working from there. `minicom` is something like `PROCOMM` for Linux. It should be present on your system if you chose to install it during your RedHat initial setup. Here is a post from a newsgroup comp.os.linux.help which explains how to start ppp manually using `minicom` (edited for space):

> From: mark <balthazaar@one.net.au> Subject: Re: pppd problem with kppp
> BachuZ wrote:
> >>Also, for an experiment, try using minicom to connect to your ISP, start ppp manually..this can prove buggy scripts. >how would u do that?
>
> Easy!! If your ISP doesn't allow a manual logon then you might be in trouble. Every ISP I've ever used does allow this, so.. 1. Start minicom. 2. dial your ISP. 3. Log in. 4. AFter your ISP starts PPP, quit minicom with ALT-Q (or whatever the sequence is to 'quit without reset'). 5. start pppd, eg:
>
> ```
> pppd -d -detach /dev/modem  115200 &
> ```
>
> OK, PPP will be running. Try pinging your ISP or another known IP address. That will test everything is OK. BTW, this is all in the PPP-HOWTO. If you can get PPP running this way, then you have a scripting problem. If PPP doesn't work, you have a PPP configuration problem. Cheers.

Command line. If you would like to start your ppp from the command line, run `netconf` (as root) to configure your first ppp interface (ppp0). The information you must enter is similar to what you entered when setting up kppp (have a look above!): the proper device for the modem port, modem initialization and dialup strings, the telephone number of your Internet Service Provider (ISP), the proper authentication protocol (by entring the login name and password into the right slot). The `netconf` utility lacked a place to enter the IP addresses of my ISP DNS server, so I edited the file `/etc/ppp/pap-secrets` (I use PAP authentication protocol) and added the two DNS IP addresses at the end of the setup line which was created by `netconf` so it looked like this:

```
# added by linuxconf
my_login_name ppp0 my_password 111.111.111.111 222.222.222.222
```

When done with the settings, I could start my ppp0 interface using the command (as root, unless I specified in `netconf` that normal users can start the interface):

```
ifup ppp0
```

and shut it down with

```
ifdown ppp0
```

Setting up the command line ppp was not more difficult on my machine than running kppp. There is lots of command line scripts to start/stop ppp, but they apparently are not so easy to set up and use, and many newbies seem to have problems with them.

If this minimalistic setup of ppp does not work for you, here are some useful links:

Roderick A. Anderson < raanders@altoplanos.net > wrote:

> I have a web page on setting up diald to work with RedHat Linux 5.x that works for me every time. It is at http://home.altoplanos.net/~raanders/diald.html

Bill Unruh < unruh@physics.ubc.ca > wrote:

> I just wanted to bring your attention to the page http://axion.physics.ubc.ca/ppp-linux.html for detailed instructions for setting up ppp. This is especially for cases in which the remote side uses (perhaps without the ISP even telling you) PAP or CHAP. While kppp is useful, there are a number of situations where it can fail.

4.5.4 How to browse the net from my networked computer without a modem?

Another computer on your network must have a modem (or another Internet connection) though :-). Set up IP masquerading. This way, all requests going from your network to your Internet Service Provider (ISP) appear to have originated from a single computer, and your ISP will let them through. Simple masquerading using RH5.2 required just one command (on the computer with the modem):

```
/sbin/ipfwadm -F -p m
```

This sets up masquerading as your default forwarding policy of your IP firewall, and therefore is insecure but probably ok for a home user. (The danger is that somebody from your network may set his route to "tunnel" back to your network thus concealing his identity.) For more info, please check the file /usr/doc/HOWTO/mini/IP-Masquerade. A more secure setup is shown here:

```
ipfwadm  -F -p deny
ipfwadm  -F -a m -S 192.168.1.1/32 -D 0.0.0.0/0
ipfwadm  -F -a m -S 192.168.1.3/32 -D 0.0.0.0/0
ipfwadm  -F -a m -S 192.168.2.0/24 -D 0.0.0.0/0
```

This sets up the default policy to "deny" and explicitly masquerades two machines with IPs 192.168.1.1 and 192.168.1.3. It also masquerades any machine from the network 192.168.2.0. The number /32 stands for point-to-point networking (this means "machine-to-machine"), the option /24 identifies a class C network. The -D 0.0.0.0/0 identifies the default route that the machines to be masqueraded use to go out to the Internet.

ON **REDHAT 6.0** THE NAME OF THE COMMAND is `ipfwadm-wrapper` (instead of `ipfwadm`) and I had to use the second, more secure method (setting up masquerading as the default policy does not seem to work on my system any more).

If you would like to have this command(s) always executed on your system startup, add it as the last line(s) to the file `/etc/rc.d/rc.local`. This file is something like AUTOEXEC.BAT in DOS. As always, it is recommended to read the manual page and other documentation to see what the command(s) does and what are the other options:

```
less /usr/doc/HOWTO/mini/IP-Masquerade
man ipfwadm
```

4.5.5 How to use Samba?

Samba (smb) is for Linux-MS Windows networking. It is a program that makes a Linux computer pretend to be a MS Windows NT server, and thus lets your MS Windows 3.1/95/98-based computers connect to the network. Samba not only replaces WinNT--it is acclaimed to do a much better job than WinNT!

One of the three machines on my home network is a dual Linux/Win95 boot. I configured my network so that if I boot Win95, another Linux machine acts as a Samba server. This way, the Win95 machine has access to the network printer, shared directories on Linux, can telnet, browse the Internet through a modem on Linux, etc. I can also access the files on the remote Win95 machine from Linux computers.

It is necessary to configure Samba only on one Linux machine.

First, I ran `setup` as root, choose "System Services" (RH6.0) or "ntsysv" (RH5.2) from the menu, and make sure that the "smb" service is enabled. (The program `setup` has help if you press F1. Just in case you were curious what the different services are for.) If smb is not listed there, maybe you did not install it during your RedHat setup? You may want to put your RedHat CD into the CDROM, mount the CD, start glint (RH5.2) or gnorpm (RH6.0) installation utility (as root) and add Samba to your system.

The second part is to configure Samba. This is relatively simple since the Samba configuration is done through a single, well commented file: `/etc/smb.config`. The minimum setup includes specifying the workgroup name. Note that if you don't fill up the "host allow" option, all hosts are allowed, which is probably ok for the home network. The other options in the example `/etc/smb.conf` supplied with your RedHat CD may require adjustment to suit your particular needs, but they didn't have to be changed to get a Windows machine connected to my Samba server.

Check that the options

```
domain master = yes
domain logons = yes
```

are enabled in `/etc/smb.conf`. You probably want them.

Under RH6.0, you may also perform a basic samba setup using (as root):

```
netconf
```

The third step is to configure your MS Windows 95 (or whatever) for networking. This is done exactly as if you were connecting to a WinNT server. Make sure to enable the networking ("client for Microsoft Network") and fill up the workgroup name. To login on the network when booting MS Windows, use your Linux user ID and password. To see if it worked, click on the icon "Network Neighborhood"--your Linux server should be listed there, and underneath you should see the shared directories and printers that you chose to share in the file `/etc/smb.conf`.

If you can't see or use the public directories, make sure that you created them and set the proper read/write permissions for all users.

MS Windows 95b and above (95c, 98, and newer NTs) may use password encryption. This will make your logins from the Window's machine fail, and you may need to enable this option in the `/etc/smb.conf` file:

```
encrypt passwords = yes
```

On the Linux server, you can start, stop, restart and check the Samba status using these commands (as root):

```
samba start
samba status
samba restart
samba stop
```

You need to re-start samba after making changes to your /etc/smb.conf file.

You can browse the net using your Netscape for Windows if you are connected to your Linux computer through Samba and the Linux machine is currently connected to the Internet. To do this, the IP masquerading must be set up on the Linux machine with the modem (described here [p 85]) and you must enable the Samba dns name resolution in `/etc/smb.conf`:

```
dns proxy = yes
```

and then tell Windows to enable the dns server, specify the Linux server name and IP address (in ControlPanel-Network-TCP/IP).

You can also mount a remote Windows directory onto your Linux filesystem. Look here [p 65] to see how.

The above described just a minimal Samba setup. You can get more information from:

```
man samba; man smb.conf; man smbclient; man smbmount
less /usr/doc/HOWTO/SMB-HOWTO    (under RH5.2)
documentation in /usr/doc/samba-2.0.3/doc    (under RH6.0)
http://www.sfu.ca/~yzhang/linux/samba/    (samba minihowto)
http://www.germanynet.de/teilnehmer/101,69082/samba.html
```

and also by studying the `file /etc/smb.conf` .

4.5.6 Sendmail

Sendmail is the Internet standard mail-transport system and the default mail-transport on RedHat5.2 and RedHat6.0 Linux. As a user, you don't use sendmail directly--sendmail is the underlying engine that manages the mail on your machine in the background. To read/send mail in the text mode, you probably want to use pine or elm (choose one and stay with it--it can be inconvenient to manage two separate mail boxes). In KDE, you may choose to use the "mail client" available from the K-menu. To communicate with the outside world through your ppp dial-out, you probably would like to use the Mozilla mailer that is built into your Netscape and which communicates directly with your remote Internet-Service-Provider-based mailbox.

Sendmail is very flexible and robust, but also notoriously difficult to manage if you needed to customize it to your specific needs. Luckily, Red Hat (5.2 or 6.0) comes with the sendmail that runs out-of-box (with some limitations though).

On my home system (default Red Hat setup), I can send mail to another user on the same machine with no problem (e.g. using pine). I can send mail locally or anywhere in the world from any mailer once I am connected to my Internet Service Provider through the modem (IP masquerading enabled [p 85]). But when I am not connected, the mail sits in the queue and waits for the Internet connection, even if the mail is to be sent to another computer on my home network (a minor annoyance). It gets sent once I connect. (It happens because sendmail is trying to do a DNS lookup and this is not available on my system--RedHat default DNS is cache-only.) If you really want to avoid this on RH6.0, you may use `netconf` (as root) and specify that sendmail is not to use DNS at all (Linuxconf that comes with RH5.2 does not give you the choice). I do use DNS.

Under Linux 6.0, you may also want to configure other sendmail options using (as root) `netconf`. I entered the name of my Internet Service Provider and the names of other machines on my home network under "mail deliver system"-"relay to hosts". I also added the names of my home network machines under "relay for by name".

You should also be aware of the limitation of your simple setup that arises if you send e-mail from your home network (for example using pine), to the world beyond your network. If you invented the IP address and your domain name is unregistred, there is no chance you will receive a reply. Your outgoing mail is given a "reply" address in the form: "user_login_name@machine.domain". This is

NOT ok since your domain name does not exist, according to any DNS in the outside world, hence no way for any reply to get to you. To overcome this, you may use the Netscape mailer to communicate to the outside of your home network. Netscape uses the settings you enter in its "edit-preferences" to communicate directly to your ISP-based mailbox (which resides on a registered server of your ISP) and thus bypasses your unregistered-home-network-based e-mail system. Another possibility is to specify the correct address in the "reply to" field. If you do it in Netscape, you can setup your NetscapeMail to use your local Linux computer as the mail server, and this way be able to send e-mail from your NetscapeMail also to the computers on your home network (not only in the "outside world"). The option "reply-to" can also be set in the KDE "mail client" setting. Also in "pine" you can specify the reply-to address under "setup-configure", "customized-hdrs" with something like:

```
Reply-to:  joe@joe_net.net
```

where "joe@joe_net.net" is your good reply-to e-mail address.

4.5.7 Simple web server (running Apache)

To set up a simple apache web server was extremely easy under RedHat 6.0. First of all, I made sure to install the apache web server rpm package during my RedHat installation. If you didn't do it, you can put the RH CD in your CDROM now, mount it, and install the package apache-*.rpm (instead of the "*" the name also contains the version number and the platform). If I know what I want to install, it is simple do it (as root):

```
cd /mnt/cdrom/RedHat/RPMS
rpm -ivh apache*.rpm
```

Then, I run "setup" (as root) and make sure that the httpd daemon (under "system services") is enabled. [A daemon is a program that sits in the background and wakes up when it is needed. In the case of httpd, it gets awakened when somebody calls on your http server.]

Now, the httpd will start automaticly every time I boot the computer. I can also start it manually using (as root):

```
/etc/rc.d/init.d/httpd start
```

and shut it down using:

```
/etc/rc.d/init.d/httpd stop
```

I should already have the directory: `/home/httpd` (check if it exists), and under it, the subdirectory `html`--this is the "root directory" for people accessing my computer from the web ("their root directory" means that they will not be able to access any directories above the directory `/home/httpd/html/` on your system). On default, this directory contains some html files and manual that the apache instalation program put there. So I should now be able to connect to my web server from another machine on my home network. For example I would type on the Netscape "Location" line:

```
http://my_http_server_name
```

and be able to browse the "apache" manual.

To put my own content on my web server, I move the "apache" sample files somewhere else, and copy or link my html files (the ones which I want to display to the public) to the directory `/home/httpd/html/` (don't forget to include the the file `index.html` , this is the one that appears first when somebody connects to your server).

The apache configuration files and log files are in the directory `/etc/html/` if you wanted to view/customize them. The log file can be viewed in "real time" using this simple command (as root):

```
tail -f /etc/httpd/logs/access_log
```

[The "tail" command normally displays the end ("tail") of a text file. With the option "-f", "tail" keeps diplaying the end of the log file as it grows--really handy to monitor the log file and see who logs onto your server.]

For graphical setup of the Apache server, try this (in an X-window, as root, if you installed "Comanche" from your RedHat CD):

```
comanche
```

4.5.8 Simple ftp server

With my RedHat Linux distribution, setting up an ftp server could not be simpler--it just works out-of-box. This is because the ftp service is enabled on default as one of the standard services (as is telnet and gopher), in the file /etc/inetd.conf. Here is the relevant part of my /etc/inted.conf:

```
ftp     stream  tcp     nowait  root    /usr/sbin/tcpd  in.ftpd -l
-a
telnet  stream  tcp     nowait  root    /usr/sbin/tcpd  in.telnetd
gopher  stream  tcp     nowait  root    /usr/sbin/tcpd  gn
```

The second part of my ftp server setup is in the file /etc/passwd which defines the ftp account:

```
ftp:*:14:50:FTP User:/home/ftp:
```

The ":" is a field separator. The first field is the user name "ftp", the "*" in the second field indicates that the password is disabled (nobody can login for this user), the user id is 14, group id is 50, "FTP User" is a comment, the home directory is /home/ftp, there is no shell specified for this user (equivalent of the shell /bin/false , meaning no shell).

Because this setup was already done for me by RedHat, anybody can ftp my computer and either login as a user (will be prompted for password and directed to his/her home directory), or login as "anonymous" and give his/her e-mail address as a password. Any user can also enter something like this on the Netscape "location" line:

```
ftp://my_computer_name
```

and connect automatically (Netscape will take care of sending the "anonymous" user name and the e-mail adddress as password).

The "anonymous" ftp users are directed to the directory `/home/ftp`, which appears to be a root directory to them (they cannot access any directory above it). I put the files I want to serve in the subdirectory `/home/ftp/pub` .

The directory /home/ftp/bin contains the commands that the remote users are able to execute. On my system these are: `compress`, `cpio`, `gzip`, `ls`, `sh`, `tar`, `zcat`; all with execute-only (111) permissions.

The directory `/home/ftp/etc` contains the setup files necessary for the anonymous account to function (edited `passwd`, `group`, `ld.so.cache`). The directory `/home/ftp/lib` contains the libraries (I guess these libraries are used by the commands that the anonymous ftp users are allowed to run).

4.5.9 How can one access my computer from the outside world when I am on the net using my phone ppp conection?

The only difficulty is that your IP address is dynamically allocated to you by your Internet Service Provider (ISP) from their IP address pool, and therefore the IP address is not the same every time you connect (unless you made specific arrangements with your ISP). To telnet, ftp, or access your web pages (served by your appache web server) from the outside world, one has to know your current IP address. To find out my current IP address, I use this "interface configuration" command which, when run without any parameters, just displays info on all active network interfaces present on your machine:

```
/sbin/ifconfig
```

On my machine this displays three paragraphs of information on: eth0 (the first ethernet network interface that leads to other computers on my home network), lo (the loopback-only interface, the one with IP 127.0.0.1, this one must be present on every machine), and ppp0 (the first point-to-point protocol interface). My current IP address, assigned to me by my ISP, is displayed under the ppp0 heading. (Your Linux machine can have multiple IP addresses assigned at the same time, so if you have a "static" IP that you use on your home network, it is still valid but visible only on your home network.)

Once I know the IP address, I can send it through ICQ or e-mail to a friend, who can then, for example, telnet my computer (s/he must have an account on my machine) and run a program on my linux machine, or enter http://my_ip_address on the "location" line in the browser to browse my home web pages, etc. I can also write a short script that will automatically notify my friend when I am connecting to the Internet and enter the name of the script under kppp-setup-account-edit-dial-"execute program upon connect". Here is my script which notifies me at work when somebody in my house is going on-line (I entered the text into a text file and made the

file executable using `chmod o+x file_name`):

```
#!/bin/bash
sleep 15
/sbin/ifconfig | mail -s notification my_email_address
```

The first line of this script tells my computer to interpret this text file as a bash shell script. The second line makes the script wait 15 seconds (just to make sure that the e-mail is not sent before the ppp connection is fully established). The third line executes the ifconfig command and pipes the output to the mail utility that sends it to my_email_address under the subject "notification".

A more flexible way to access your home computer remotly is to configure it as a dial-up ppp server (as oposed to the dial-in client that you use when you connect to your ISP). If somebody has a simple recipe how to do it, please drop a line.

4.5.10 Can my home computer get hacked?

Unfortunately, this is perfectly possible and attempts to do so are quite common. Every time you are connected to your Internet Service Provider (ISP) you are at risk. Read the previous answer if you would like to know how it is possible.

The real danger is that the intruder, if s/he is able to login onto your machine on any account, might find another security hole and get root access. This is particularly possible if you are a real newbie administrator and/or your machine is not really security oriented (you are at home, aren't you--who would care about security, you think!).

To protect yourself, have fairly long passwords that contain both numbers and letters for all accounts on your computer. Change the passwords occasionally. The best way to enforce the password policies on all users of your computer is to run (as root, available on RH6.0) `linuxconf` and under "password and account policies" change the minimum password length to 6 or more characters, the minimum number of non-alphanumeric characters (i.e., not-letters) to 1 or 2, the number of days after which the password must be changed to something like 90 or less, and set a warning about password expiry to something like 7 days before the expiration. Check here for more info on weak passwords.

It is also an excellent idea to ocassionally screen the files that contain a record of all the logins onto your computer: `/var/log/secure` (the most recent log) `/var/log/secure.1` (older log) `/var/log/secure.2` (yet older log), etc. There are also other useful log files in the directory `/var/log` that you might want to view, check them out from time to time.

If you never use remote connections to your home Linux machine, you might want to restrict the rights to use telnet, ftp, etc. (all the network services are listed in the file `/etc/inetd.conf`) to the machines on your home network. The access is controlled by two files: `/etc/hosts.allow` and `/etc/hosts.deny` . These access-control files work as follows. When an outside connection is requested, the file `/etc/host.allow` is scanned first and if the name of the machine from which the connection is requested is matched, the access is granted (irrespectively of any entry in `/etc/host.deny`). Otherwise, the file `/etc/host.deny` is scanned, and if the

name of the machine from which the connection is requested is matched, the connection is closed. If no matches are found in either file, the permission is granted. As an example, you can deny access to telnet and ftp your home server from any machine from outside your home network by inserting the following entry in the file `/etc/hosts.deny` :

```
in.telnetd, in.ftpd: ALL EXCEPT LOCAL, .your_home_domain.name
```

For more info, check the excellent "Linux Network Administrator Guide" which is surely present on your RedHat (or whatever) distribution CD. I printed this book and had it hardcovered.

Go to part 5: Kernel Upgrade

LINUX NEWBIE ADMINISTRATOR GUIDE

ver. 0.54 1999-10-15

Part 5: Kernel Upgrade

Author: Alesh Mustar (alesh@jpdesign.net)

Answers to Some Frequently Asked Linux Questions

Distributed under the General Public Licence http://www.gnu.org/copyleft/gpl.html. Your feedback, comments, corrections, and improvements are appreciated. Comment specific to this page: alesh@jpdesign.net. Comment on the balance of this guide: bklimas@magma.ca

---

Quick site navigation:

Start: Linux Newbie Administrator Guide
Part 0: For the Undecided (Linux Benefits)
Part 1: Before Linux Installation
Part 2: Linux Resources, Help and Some Links
Part 3: Basic Operations FAQ
Part 4: Linux Newbie Administrator FAQ
Part 5: >How to Upgrade the Kernel<
Part 6: Linux Shortcuts and Commands
Part 7: Essential Linux applications (proprietary or not)

Kernel Upgrade - version 1.1.0
October 11th 1999
Author: Alesh Mustar (alesh@jpdesign.net)

Contents of this page:

5. How to upgrade your kernel?

5.1 Introduction
5.2 What is kernel?
5.3 Why upgrade?
5.4 Where can I get the new kernel?
5.5 Should I download full source or the patch only?
5.6 Unpacking the download file
5.6.1 Unpacking the patch
5.6.2 Unpacking the full source
5.7 Compiling new kernel
5.7.1 Using the patch
5.7.2 Using the full source
5.8 Installing the new kernel

5.9 What about next kernel upgrade?
5.10 Notice for Red Hat users
5.11 About the author
5.12 Changes

---

5. How to upgrade your kernel?

5.1 Introduction

This document was written to help the new users of Linux upgrade their kernel.
I used various sources for writting this document. From the official HOW-TOs (http://metalab.unc.edu/LDP/HOWTO/) to the Linux Knowledge Base and other sources.

5.2 What is kernel?

The kernel acts as a mediator for your programs and your hardware. First, it does (or arranges for) the memory management for all of the running programs (processes), and makes sure that they all get a fair (or unfair, if you please) share of the processor's cycles. In addition, it provides a nice, fairly portable interface for programs to talk to your hardware.
There is certainly more to the kernel's operation than this, but these basic functions are the most important to know.

5.3 Why upgrade?

There are serveral reasons to upgrade your kernel. Most of the reasons are that you perhaps need a new device driver for your new or old hardware (the previous device driver could have had a bug somewhere) or you need to upgrade due to a bug in the kernel (security one or a non-security one). Usually new kernels run faster then the old ones and are more stable and reliable.

5.4 Where can I get the new kernel?

First you need to get the new kernel. Load a browser (any kind of) and point it to http://www.kernel.org/ . There you can find a list of mirrors. Mirrors are sites with the (mostly) same information as the main site. Since the main kernel.org site is heavily overloaded with downloads choose a mirror site closest to you or even the one located in your country. For example: to get the new kernel from Norway point your browser to the .no kernel archive => http://www.no.kernel.org/ . Notice the .no after the www string. Lots of countries have mirrors for the kernel archive.
Once the mirror page loads you will find information on that page which looks like this:

<= start of example page=>

The Linux Kernel Archives mirror at Sunsite.uio.no
All transfers are logged. If you don't like this policy please disconnect now.

Welcome to the Linux Kernel Archives. This is an official mirror site for the Linux kernel source. Please see `http://www.kernel.org/mirrors/` for information about how to connect to an other kernel-mirror site.

Protocol	URL
HTTP	http://linux-kernel.uio.no/pub/
FTP	ftp://linux-kernel.uio.no/pub

What is Linux?Linux is a Unix clone written from scratch by Linus Torvalds with assistance from a loosely-knit team of hackers across the Net. It aims towards POSIX compliance.
...
<= end of example page=>

Now choose a protocol which suits you better to transfer the files. Lets say we will choose HTTP protocol (same protocol as your WWW browser uses).
You browser will display a page with a directory structure, perhaps similar to this:

<= start of example page =>

Index of /pub

Name Last modified Size Description

[DIR] Parent Directory 28-Dec-98 12:06 -
[DIR] linux/ 31-Dec-98 09:57 -
[ ] ls-lR 29-Mar-99 14:56 262k
[DIR] software/ 31-Dec-98 18:25 -

<= end of example page =>

Note the *linux* directory. Point your browser at it. A page like this will show:

<= start of example page =>

Index of /pub/linux

Name Last modified Size Description

[DIR] Parent Directory 29-Mar-99 14:56 -
[DIR] daemons/ 12-Feb-99 01:19 -
[DIR] devel/ 18-Dec-98 19:34 -
[DIR] distributions/ 18-Dec-98 19:34 -
[DIR] docs/ 31-Jan-99 04:01 -
[DIR] kernel/ 18-Mar-99 21:55 -
[DIR] libs/ 18-Dec-98 19:35 -
[DIR] utils/ 31-Jan-99 04:01 -

<= end of example page =>

This site contains lots of things to download but currently we are only interested in the kernel. Point your browser at the *kernel* directory.
Something like this should show:

<= start of example page =>

Index of /pub/linux/kernel

Name Last modified Size Description

[DIR] Parent Directory 31-Dec-98 09:57 -
[ ] COPYING 13-Mar-94 00:00 18k
[ ] CREDITS 16-Sep-96 00:00 36k
[DIR] Historic/ 18-Dec-98 20:55 -
[DIR] SillySounds/ 18-Dec-96 09:45 -
[DIR] alan/ 18-Mar-99 21:55 -
[DIR] davem/ 29-Jan-99 04:05 -
[DIR] hpa/ 19-Dec-98 02:16 -
[DIR] people/ 18-Mar-99 21:55 -
[DIR] testing/ 27-Mar-99 00:55 -
[DIR] v1.0/ 19-Dec-98 02:19 -
[DIR] v1.1/ 19-Dec-98 03:51 -
[DIR] v1.2/ 19-Dec-98 04:36 -
[DIR] v1.3/ 19-Dec-98 13:21 -
[DIR] v2.0/ 11-Feb-99 16:48 -
[DIR] v2.1/ 11-Feb-99 16:45 -
[DIR] v2.2/ 24-Mar-99 00:27 -
[DIR] whawes/ 18-Dec-98 19:34 -

Linux kernel release 2.0.xx

These are the release notes for linux version 2.0. Read them carefully,
as they tell you what this is all about, explain how to install the
kernel, and what to do if something goes wrong.
...
....

<= end of example page =>

For our example we will take in consideration that while writting this the kernel version 2.2.x series is the stable version. That’s why we choose the *v2.2/* directory on the server.
How do you know which is the latest version? At the bottom of the http://www.kernel.org/ page you have a text like this:

<= start of example page =>

The latest stable version of the Linux kernel is:	2.2.12
The latest beta version of the Linux kernel is:	2.3.20
The latest prepatch (alpha) version *appears* to be:	none

<= end of example page =>

That is your information to find out which is the latest stable kernel and which one is also recomended to use. Also as you will see you can ussualy see a file with the name like *LATEST-IS-2.2.12* at the top of the directory listing which tells you the latest version.

Once that we are in the *v2.2/* directory on our server we will see something like this:

<= start of example page =>

Index of /pub/linux/kernel/v2.2

Name Last modified Size Description

[DIR] Parent Directory 16-Aug-99 09:13 -
[ ] LATEST-IS-2.2.12 26-Aug-99 02:45 0k
[ ] linux-2.2.0.tar.bz2 26-Jan-99 02:41 10.1M
[ ] linux-2.2.0.tar.bz2.s..26-Jan-99 02:41 1k
[CMP] linux-2.2.0.tar.gz 26-Jan-99 02:41 12.5M
[ ] linux-2.2.0.tar.gz.sign26-Jan-99 02:41 1k
[ ] linux-2.2.1.tar.bz2 28-Jan-99 21:56 10.1M
[ ] linux-2.2.1.tar.bz2.s..28-Jan-99 21:56 1k
[CMP] linux-2.2.1.tar.gz 28-Jan-99 21:56 12.5M
[ ] linux-2.2.1.tar.gz.sign28-Jan-99 21:56 1k
[ ] linux-2.2.10.tar.bz2 14-Jun-99 07:33 10.8M
[ ] linux-2.2.10.tar.bz2...14-Jun-99 07:33 1k
[CMP] linux-2.2.10.tar.gz 14-Jun-99 07:33 13.3M
[ ] linux-2.2.10.tar.gz.s..14-Jun-99 07:33 1k
[ ] linux-2.2.11.tar.bz2 10-Aug-99 01:03 11.2M
[ ] linux-2.2.11.tar.bz2...10-Aug-99 01:03 1k
[CMP] linux-2.2.11.tar.gz 10-Aug-99 01:03 13.8M
[ ] linux-2.2.11.tar.gz.s..10-Aug-99 01:03 1k
[ ] linux-2.2.12.tar.bz2 26-Aug-99 02:45 11.5M
[ ] linux-2.2.12.tar.bz2...26-Aug-99 02:45 1k
[CMP] linux-2.2.12.tar.gz 26-Aug-99 02:45 14.2M
[ ] linux-2.2.12.tar.gz.s..26-Aug-99 02:45 1k
[ ] linux-2.2.2.tar.bz2 23-Feb-99 03:58 10.1M
[ ] linux-2.2.2.tar.bz2.s..23-Feb-99 03:58 1k
[CMP] linux-2.2.2.tar.gz 23-Feb-99 03:58 12.5M
[ ] linux-2.2.2.tar.gz.sign23-Feb-99 03:58 1k
[ ] linux-2.2.3.tar.bz2 09-Mar-99 01:42 10.2M
[ ] linux-2.2.3.tar.bz2.s..09-Mar-99 01:42 1k
[CMP] linux-2.2.3.tar.gz 09-Mar-99 01:42 12.6M
[ ] linux-2.2.3.tar.gz.sign09-Mar-99 01:42 1k
[ ] linux-2.2.4.tar.bz2 23-Mar-99 23:33 10.4M
[ ] linux-2.2.4.tar.bz2.s..23-Mar-99 23:33 1k
[CMP] linux-2.2.4.tar.gz 23-Mar-99 23:33 12.8M
[ ] linux-2.2.4.tar.gz.sign23-Mar-99 23:33 1k
[ ] linux-2.2.5.tar.bz2 29-Mar-99 08:54 10.4M
[ ] linux-2.2.5.tar.bz2.s..29-Mar-99 08:54 1k
[CMP] linux-2.2.5.tar.gz 29-Mar-99 08:54 12.9M
[ ] linux-2.2.5.tar.gz.sign29-Mar-99 08:54 1k
[ ] linux-2.2.6.tar.bz2 16-Apr-99 23:46 10.5M

[ ] linux-2.2.6.tar.bz2.s..16-Apr-99 23:46 1k
[CMP] linux-2.2.6.tar.gz 16-Apr-99 23:46 13.0M
[ ] linux-2.2.6.tar.gz.sign16-Apr-99 23:46 1k
[ ] linux-2.2.7.tar.bz2 28-Apr-99 20:42 10.6M
[ ] linux-2.2.7.tar.bz2.s..28-Apr-99 20:42 1k
[CMP] linux-2.2.7.tar.gz 28-Apr-99 20:42 13.0M
[ ] linux-2.2.7.tar.gz.sign28-Apr-99 20:42 1k
[ ] linux-2.2.8.tar.bz2 11-May-99 21:59 10.7M
[ ] linux-2.2.8.tar.bz2.s..11-May-99 21:59 1k
[CMP] linux-2.2.8.tar.gz 11-May-99 21:59 13.2M
[ ] linux-2.2.8.tar.gz.sign11-May-99 21:59 1k
[ ] linux-2.2.9.tar.bz2 14-May-99 01:54 10.7M
[ ] linux-2.2.9.tar.bz2.s..14-May-99 01:54 1k
[DIR] Parent Directory 16-Aug-99 09:13 -
[ ] LATEST-IS-2.2.12 26-Aug-99 02:45 0k
[ ] linux-2.2.0.tar.bz2 26-Jan-99 02:41 10.1M
[ ] linux-2.2.0.tar.bz2.s..26-Jan-99 02:41 1k
[CMP] linux-2.2.0.tar.gz 26-Jan-99 02:41 12.5M
[ ] linux-2.2.0.tar.gz.sign26-Jan-99 02:41 1k
[ ] linux-2.2.1.tar.bz2 28-Jan-99 21:56 10.1M
[ ] linux-2.2.1.tar.bz2.s..28-Jan-99 21:56 1k
[CMP] linux-2.2.1.tar.gz 28-Jan-99 21:56 12.5M
[ ] linux-2.2.1.tar.gz.sign28-Jan-99 21:56 1k
[ ] linux-2.2.10.tar.bz2 14-Jun-99 07:33 10.8M
[ ] linux-2.2.10.tar.bz2...14-Jun-99 07:33 1k
[CMP] linux-2.2.10.tar.gz 14-Jun-99 07:33 13.3M
[ ] linux-2.2.10.tar.gz.s..14-Jun-99 07:33 1k
[ ] linux-2.2.11.tar.bz2 10-Aug-99 01:03 11.2M
[ ] linux-2.2.11.tar.bz2...10-Aug-99 01:03 1k
[CMP] linux-2.2.11.tar.gz 10-Aug-99 01:03 13.8M
[ ] linux-2.2.11.tar.gz.s..10-Aug-99 01:03 1k
[ ] linux-2.2.12.tar.bz2 26-Aug-99 02:45 11.5M
[ ] linux-2.2.12.tar.bz2...26-Aug-99 02:45 1k
[CMP] linux-2.2.12.tar.gz 26-Aug-99 02:45 14.2M
[ ] linux-2.2.12.tar.gz.s..26-Aug-99 02:45 1k
[ ] linux-2.2.2.tar.bz2 23-Feb-99 03:58 10.1M
[ ] linux-2.2.2.tar.bz2.s..23-Feb-99 03:58 1k
[CMP] linux-2.2.2.tar.gz 23-Feb-99 03:58 12.5M
[ ] linux-2.2.2.tar.gz.sign23-Feb-99 03:58 1k
[ ] linux-2.2.3.tar.bz2 09-Mar-99 01:42 10.2M
[ ] linux-2.2.3.tar.bz2.s..09-Mar-99 01:42 1k
[CMP] linux-2.2.3.tar.gz 09-Mar-99 01:42 12.6M
[ ] linux-2.2.3.tar.gz.sign09-Mar-99 01:42 1k
[ ] linux-2.2.4.tar.bz2 23-Mar-99 23:33 10.4M
[ ] linux-2.2.4.tar.bz2.s..23-Mar-99 23:33 1k
[CMP] linux-2.2.4.tar.gz 23-Mar-99 23:33 12.8M
[ ] linux-2.2.4.tar.gz.sign23-Mar-99 23:33 1k
[ ] linux-2.2.5.tar.bz2 29-Mar-99 08:54 10.4M
[ ] linux-2.2.5.tar.bz2.s..29-Mar-99 08:54 1k
[CMP] linux-2.2.5.tar.gz 29-Mar-99 08:54 12.9M
[ ] linux-2.2.5.tar.gz.sign29-Mar-99 08:54 1k
[ ] linux-2.2.6.tar.bz2 16-Apr-99 23:46 10.5M
[ ] linux-2.2.6.tar.bz2.s..16-Apr-99 23:46 1k
[CMP] linux-2.2.6.tar.gz 16-Apr-99 23:46 13.0M
[ ] linux-2.2.6.tar.gz.sign16-Apr-99 23:46 1k
[ ] linux-2.2.7.tar.bz2 28-Apr-99 20:42 10.6M
[ ] linux-2.2.7.tar.bz2.s..28-Apr-99 20:42 1k
[CMP] linux-2.2.7.tar.gz 28-Apr-99 20:42 13.0M
[ ] linux-2.2.7.tar.gz.sign28-Apr-99 20:42 1k
[ ] linux-2.2.8.tar.bz2 11-May-99 21:59 10.7M
[ ] linux-2.2.8.tar.bz2.s..11-May-99 21:59 1k
[CMP] linux-2.2.8.tar.gz 11-May-99 21:59 13.2M
[ ] linux-2.2.8.tar.gz.sign11-May-99 21:59 1k
[ ] linux-2.2.9.tar.bz2 14-May-99 01:54 10.7M
[ ] linux-2.2.9.tar.bz2.s..14-May-99 01:54 1k

<= end of example page =>

5.5 Should I download full source or the patch only?

Now you have a choice to download either the patch or the full source.
If you're upgrading from 2.0.x kernel series you will have to download the full source.
If you're upgrading a 2.2.x kernel series you can download only the patch, which is ofcourse much more smaller then the full source. The patch containts only the differences.
If you would like more detailed info on patches and diff take a look at this link:
http://www.linuxpower.org/display_item.phtml?id=101 .
Now grab *your choice* and download it. I highly suggest you download it to */usr/src/* directory.
I suggest downloading the file with the *.tar.gz* extension (if you're downloading the full source). For patches I suggest *.bz2* extension. Please note that the examples which follow will be based on this decisions (about file types).

5.6 Unpacking the download file

5.6.1 Unpacking the patch

We are assuming that you downloaded the patch into */usr/src/* directory. For our example lets say the patch filename is *patch-2.2.12.bz2*. Before using it we must first unpack it. *.bz2* extension is bunzip2 format. By using the following command we unpack the file:

<= start of example command =>

bunzip2 patch-2.2.12.bz2

<= end of example command =>

No visual output will be shown. Check what files are located in the directory now. You will notice that the *patch-2.2.12.bz2* file is gone and *patch-2.2.12* is there of a much bigger size.

5.6.2 Unpacking the full source

Again we are assuming that you downloaded the full source into */usr/src/* directory. For our example lets say the source filename is *linux-2.2.12.tar.gz*.
Before even upacking it we must first do something else. In */usr/src/* you have a symbolic link named *linux* which points to for example at *linux-2.0.36*. How can you check that out? Type *ls -al* in the */usr/src/* directory. You'll see something like this:

<= start of example page =>

```
total 7295
drwxr-xr-x 5 root root 1024 Mar 19 15:01 .
drwxr-xr-x 20 root root 1024 Mar 13 22:52 ..
lrwxrwxrwx 1 root root 11 Mar 15 18:21 linux -> linux-2.0.36
drwxr-xr-x 17 root root 1024 Mar 13 22:52 linux-2.0.36
...
```

<= end of example page =>

Notice the *lrwxrwxrwx 1 root root 11 Mar 15 18:21 linux -> linux-2.0.36* which is the symbolic link pointing to *linux-2.0.36* directory.
If we would unpack the kernel source now, by default it would be unpacked into the *linux* directory overwriting our previous source.
Lets delete the link by typing *rm linux*. It only removes the link, not the directory!
Now lets unpack the source. Type *tar -xvfz linux-2.2.12.tar.gz*. Lots of text will fly by. If you do a *ls -al* now you'll notice the *linux* directory. That is our new kernel source directory.

5.7 Compiling new kernel

5.7.1 Using the patch

If you downloaded the patch you will need to go thru this section.
We are assuming right now that you are located in */usr/src/* directory. We have unpacked the patch (Section 5.6.1). Now we will move to the kernel directory (assuming *linux*). Type *cd linux*.
Now we are located in the kernel directory. At this point we have a lot of options and questions. Is this your first patch of the kernel? No? Yes? We won't go into details and we will assume that this is your first kernel patch. Now we are ready to patch the kernel.
To do this we will use the patch utility. For more detail and wider information about this I suggest checking out the article from the www.linuxpower.org web site, located at this url: http://www.linuxpower.org/display_item.phtml?id=101 .
Type *patch -E -p1 < ../patch-2.2.12*
Lots of text will fly by. If it doesn't stop anywhere but only at the end (otherwise it will be askign you a question) everything went ok. If not, then you might have a problem (you already patched a kernel once, or something went really really wrong).
If everything went ok read on. Now it's time to configure the kernel and then compile it. We now have 2 choices. One is to compile the new kernel totally from scratch, the other one is to just re-compile it using the old configuration. Since you used the patch we are assuming that you're gonna compile with the old configuration. If you don't wish to, read section 6.2 where we compile the kernel from scratch and use the steps there. The only difference is in step 1, where you should type *make menuconfig*.
This is the re-compile of the kernel using the old configuration. Follow these steps:
1. type *make oldconfig*
At this point you could have been asked for new drivers and development ones. Unless you really need them answer no (*n*).
2. type *make dep*
3. type *make clean*
4. type *make zImage* (if this fails use *bzImage*, which stands for *big zImage*)
5. type *make modules* (if you enabled them)
6. type *make modules_install* (if you typed #5)

If everything went ok, we have a new kernel, ready to be installed and used on the system.

5.7.2 Using the full source

If you downloaded the full source you will need to go thru this section.
We are assuming right now that you are located in */usr/src/* directory. We have unpacked the new kernel (Section 5.6.2). Now we will move to the kernel directory (assuming *linux*). Type *cd linux*.
Now we are located in the kernel directory. Now lets start our fresh compile.
1. type *make menuconfig*
Here you will have to configure your kernel. We won't go into details or explains here since the configuration is different from machine to machine. Go thru the configuration wisely and if you don't know what something is, rather leave it in then putting it out of the configuration.
2. type *make dep*
3. type *make clean*
4. type *make zImage* (if this fails use *bzImage*, which stands for *big zImage*)
5. type *make modules* (if you enabled them)
6. type *make modules_install* (if you typed #5)

If everything went ok, we have a new kernel, ready to be installed and used on the system.

5.8 Installing the new kernel

Installing the new kernel takes a knowledge of editing the *lilo.conf* file located in the */etc/* directory. I will state now that *lilo.conf*deferes from machine to machine and it might not look the same on your machine as it does on my machines.
First of all, we will assume that you are still in the */usr/src/linux/* directory. We will have to copy the *new kernel* and *System.map* to certain location which again is different on some distributions.
Lets change directory to the new kernel.
Type cd /arch/i386/boot/
Type ls -al just ot see what's there. You'll see something like this:

<= start of example page =>

```
total 588
drwxr-xr-x   4 1046     1046         1024 Apr  5 20:54 .
drwxr-xr-x   7 1046     1046         1024 Mar 13 02:39 ..
-rw-r--r--   1 1046     1046         2633 Jan  2 19:27 Makefile
-rwxr-xr-x   1 root     root          544 Apr  5 20:54 bootsect
-rw-r--r--   1 1046     1046         9536 Jun 24  1998 bootsect.S
-rw-r--r--   1 root     root         1238 Apr  5 20:54 bootsect.o
-rw-r--r--   1 root     root         8293 Apr  5 20:54 bootsect.s
drwxr-xr-x   2 1046     1046         1024 Apr  5 20:54 compressed
-rw-r--r--   1 1046     1046          904 Jan  3  1995 install.sh
-rwxr-xr-x   1 root     root         1428 Apr  5 20:54 setup
-rw-r--r--   1 1046     1046        20136 Nov 29 02:18 setup.S
-rw-r--r--   1 root     root         3053 Apr  5 20:54 setup.o
-rw-r--r--   1 root     root        24136 Apr  5 20:54 setup.s
drwxr-xr-x   2 1046     1046         1024 Apr  5 20:54 tools
-rw-r--r--   1 1046     1046        36836 Sep 30  1998 video.S
-rw-r--r--   1 root     root       477285 Apr  5 20:54 zImage
```

<= end of example page =>

If you at section 5.7.x used *make zImage* you should copy the *zImage* file then, otherwise copy the *bzImage* file. We will assume z*Image*.
The option to which we encounter right now is where to copy the kernel. On Red Hat distributions the kernel is located in */boot/* on Slackware in / . I'm not very familiar with other distributions (Debian, SuSe, etc.). Depending on your distribution make your choice where you will copy the new kernel.
We will make an example for Red Hat.
Type *cp zImage /boot/vmlinuz-2.2.12*

Now change the directory back to */usr/src/linux/* .
Type *cd /usr/src/linux/* .

Now lets copy the new *System.map*. Please note that *System.map* is always located in */boot/* .
Type *cp System.map /boot/System.map-2.2.12*

Now we have to make a new s*ymbolic link* which points to the new *System.map*.
First change the directory to */boot* by typing *cd /boot* .
Now delete the old symbolic link (note that deleting the symbolic link only deletes the link, not also the file or directory to which it points). Type *rm System.map* .
Now lets create a new symbolic link which points to our new *System.map-2.2.12*.Type *ln -s System.map-2.2.12 System.map* .

All we have to do now is to edit the *lilo.conf* file, add our new kernel in there and run *lilo*.
Lets change directory to /etc by typing *cd /etc* . With your favourite editor (vi, joe, pico) edit the *lilo.conf* file. For our example we will use *joe*.
You will see something like this in your file :

= start of example page =>

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
image=/boot/vmlinuz-2.0.36-0.7
      label=linux.old
      root=/dev/hda1
      read-only
```

<= end of example page =>

Please note that this can and probably is different on your system.
Now we will add the new kernel to our *lilo.conf* file. Copy the part from image to the end and paste it before the current image. You should get something like this:

= start of example page =>

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
image=/boot/vmlinuz-2.0.36-0.7
      label=linux.old
      root=/dev/hda1
      read-only
image=/boot/vmlinuz-2.0.36-0.7
      label=linux.old
      root=/dev/hda1
      read-only
```

<= end of example page =>

Now change the pasted part to the settings and version(s) of your new kernel. For our example (we use the 2.2.4 kernel) it looks like this:

= start of example page =>

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
image=/boot/vmlinuz-2.2.12
      label=linux
      root=/dev/hda1
      read-only
image=/boot/vmlinuz-2.0.36-0.7
      label=linux.old
      root=/dev/hda1
      read-only
```

<= end of example page =>

We changed the label part, the image part, everything else stayed that same.
I won't explain the basics of the lilo.conf file, since it's very well described in the LILO How-To .

Now run *lilo* by simply typing that. You should see something like this (more or less exactly this):

= start of example page =>

```
Added linux *
Added linux.old
```

<= end of example page =>

You can now reboot your machine to the new kernel.

5.9 What about next kernel upgrade?

Well everything should be the same only one thing not. If you will be patching the kernel you will first have to remove the old patch before applying the new one. This is done by first typing *patch -R -p1 < ../patchfile* (where patchfile is the name of the old patch) and then applying the new patch with the -E switch. Everything else should be the same.

5.10 Notice to Red Hat users

Red Hat users should note this. If you're upgrading your kernel from the 2.0.xx series to 2.2.x series you must read this. The document states similar to this one, step by step how to upgrade certain system specific things like init, samba, etc.. which for such upgrade is neccesary.
If there are similar issues for other distributions, please let me know. It would be nice if you could point me to a certain document with information about it.

5.11 About the author

My name is Alesh Mustar, I'm 20 years old, living in Slovenia, Europe working as a programmer and system administrator. Beside spending lots of time behind the computer screen(s) I like spending time with my girlfriend Tanja, reading books and driving around with my car. I can be contacted at the email address which is at the top of the page. Comments, suggestions or anything else is more then just welcome.

5.12 Changes

Version 1.1.0 - minor changes, some typo errors fixed and some kernel numbers repairs

Go to Part 6: Linux Shortcuts and Commands

LINUX NEWBIE ADMINISTRATOR GUIDE

ver. 0.59 1999-11-05

Part 6: Linux Shortcuts and Commands

by Stan and Peter Klimas

Easy Answers to Questions Frequently Asked by the New Linux Users who Install Linux on their Home Computers or Administer their Home Network

Distributed under the General Public Licence http://www.gnu.org/copyleft/gpl.html. Your feedback, comments, corrections, and improvements are appreciated. Send them to bklimas@magma.ca

---

Quick site navigation:

Start: Linux Newbie Administrator Guide
Part 0: For the Undecided (Linux Benefits)
Part 1: Before Linux Installation
Part 2: Linux Resources, Help and Some Links
Part 3: Basic Operations FAQ
Part 4: Linux Newbie Administrator FAQ
Part 5: How to Upgrade the Kernel
Part 6: >Linux Shortcuts and Commands<
Part 7: Essential Linux applications (proprietary or not)

Contents of this page:

Linux shortcuts and commands:

6.1 Linux essential shortcuts and sanity commands [p 107]
6.2 Common LINUX commands--system info [p 109]
6.3 Basic operations [p 110]
6.4 Process control [p 112]
6.5 Basic administration commands [p 113]
6.6 Networking tools [p 116]
6.7 Linux built-in developer tools and programming languages [p 117]

Linux Shortcuts and Commands:

Legend:
<> = single special or function key on the keyboard, for example <Ctrl>
italic = name of the file or variable you probably want to substitute with your own
`fixed width` = in-line Linux commands and filenames
The command is always a single line (even if it wraps in the table) unless otherwise noted.

Notes for the UNIX Clueless:
1. LINUX IS CASE-SENSITIVE. For example: Netscape, NETSCAPE and nEtscape are three different commands. Also my_filE, my_file, and my_FILE are three different files. Your user login name and password are also case sensitive. (This goes with the tradition of UNIX and the "c" programming language being case sensitive.)
2. Filenames can be up to 256 characters long and can contain letters, numbers, "." (dot), "_" (underscore), "-" (dash), plus some other not recommeded characters.
3. Files with names starting with "." are normally not shown by the `ls` (list) or dir commands. Think of these files as "hidden". Use `ls -a` (list with the option "all") to see these files.
4. "/" is an equivalent to DOS "\" (root directory, meaning the parent of all other directories).
5. Under Linux, all directories appear under a single directory tree (there are no DOS-style drive letters).
6. In a configuration file, a line starting with # is a comment.

6.1 Linux essential shortcuts and sanity commands

<Ctrl><Alt><F1>	Switch to the first text terminal. Under Linux you can have several terminals opened at the same time.
<Ctrl><Alt><Fn>	(n=1..6) Switch to the nth text terminal.
<Ctrl><Alt><F7>	Switch to the first GUI terminal (if one is running).
<Ctrl><Alt><Fn>	(n=7..12) Switch to the nth GUI terminal (if a GUI terminal is running on screen n-1)
<Tab>	Autocomplete the command (in a terminal). THIS SHORTCUT IS GREAT! It even works at LILO prompt!
<ArrowUp>	Scroll and edit the command history. Press [Enter] to execute.
<Shift><PgUp>	Scroll terminal output up. Work also at the login prompt, so you can scroll through your bootup messages.
<Shift><PgDown>	Scroll terminal output down.
<Ctrl><Alt><+>	(in X-windows) Change to the next X-server resolution (if you set up the X-server to more than one resolution). For multiple resulutions on my standard SVGA card/monitor, I have the following line in the file `/etc/X11/XFree86Config` (the first resolution starts on default, the largest determines the size of the "virtual screen"): `Modes "1024x768" "800x600" "640x480" "512x384" "480x300" "400x300" "1152x864"`
<Ctrl><Alt><->	(in X-windows) Change to the previous X-server resolution.
<Ctrl><Alt><BkSpc>	(in X-windows) Kill the current X-windows server. Use if the X-windows server crushes and cannot be exited normally.
<Ctrl><Alt><Del>	Shut down the system and reboot. This is the normal shutdown command for a user at the text-mode console. Don't just press the "reset" button for shutdown!
<Ctrl>c	Kill the current process.
<Ctrl>d	Logout from the current terminal.
<Ctrl>d	Send [End-of-File] to the current process. Don't press it twice else you also log out (see the previous command).
<Ctrl>s	Stop the transfer to the terminal.
<Ctrl>q	Resume the transfer to the terminal. Try if your terminal mysteriously stops responding.
<Ctrl>z	Send the current process to the background.
reset	Restore a screwed-up terminal (a terminal showing funny characters) to default setting. Use if you tried to "cat" a binary file. You may not be able to see the command as you type it.
<MiddleMouseButton>	Paste the text which is currently highlighted somewhere else. This is the normal "copy-paste" operation in Linux. (It doesn't work with Netscape and WordPerfect which use the MS Windows-style "copy-paste".) Best used with a Linux-ready 3-button mouse (Logitech or similar).
~	(tilde) My home directory. For example, `cd ~/my_dir` will change my working directory to the subdirectory "*my_dir*" under my home directory. Typing just "cd" alone is an equivalent of the command "cd ~".
.	(dot) Current directory. For example, `./my_program` will atempt to execute the file "my_program" located in your current working directory.
..	(two dots) Directory parent to the current one.

6.2 Common Linux commands--system info

Command	Description
pwd	Print working directory, i.e., display the name of my current directory on the screen.
hostname	Print the name of the local host (the machine on which you are working). Use `netconf` (as root) to change the name of the machine.
whoami	Print my login name.
id *username*	Print user id (uid) and his/her group id (gid), effective id (if different than the real id) and the supplementary groups.
date	Print or change the operating system date and time. E.g., change the date and time to 2000-12-31 23:57 using this command: `date 123123572000` To set the hardware clock from the system clock, use the command (as root) `setclock`
time	Determine the amount of time that it takes for a process to complete + other info. Don't confuse it with date command. E.g. I can find out how long it takes to display a directory content using: `time ls`
who	Determine the users logged on the machine.
rwho -a	Determine all users logged on your network. The rwho service must be enabled for this command to run. If it isn't, run `setup` as root to enable "rwho".
finger *user_name*	System info about a user. Try: `finger root`
last	Show listing of users last logged-in on your system.
history | more	Show the last (1000 or so) commands executed from the command line on the current account. The "| more" causes the display to stop after each screenful.
uptime	Amount of time since last reboot.
ps	(=print status) List the processes currently run by the current user.
ps axu | more	List all the processes currently running, even those without the controllling terminal, together with the name of the user that owns each process.
top	Keep listing the currently running processes, sorted by cpu usage (top users first). In KDE, you can get GUI-based Ktop from "K"menu under "System"-"Task Manager" (or by executing "ktop" in an X-terminal).
uname -a	Info on your server.
free	Memory info (in kilobytes).
df -h	(=disk free) Print disk info about all the filesystems (in human-readable form)
du / -bh | more	(=disk usage) Print detailed disk usage for each subdirectory starting at root (in human legible form).
cat /proc/cpuinfo	Cpu info. Note that the files in the /proc directory are not real files. They are hooks to look at information available to the kernel.
cat /proc/interrupts	List the interrupts in use.
cat /proc/version	Linux version and other info
cat /proc/filesystems	Show the types of filesystems currently in use.
cat /etc/printcap	Show the setup of printers.
lsmod	(as root. Use `/sbin/lsmod` to execute this command when you are a non-root user.) Show the kernel modules currently loaded.

set|more	Show the current user environment.
echo $PATH	Show the content of the environment variable "PATH". This command can be used to show other environment variables as well. Use "set" to see the full environment.
dmesg | less	Print kernel messages (the current contnent of the so-called kernel ring buffer). Press "q" to quit "less". Use `less /var/log/dmesg` to see what "dmesg" dumped into the file right after bootup.

6.3 Basic operations

any_command --help |more	Display a brief help on a command (works with most commands). "--help" works similar to DOS "/h" switch. The "more" pipe is needed if the output is longer than one screen.
ls	List the content of the current directory. Under Linux, the command "dir" is an alias to ls. Many users have "ls" to be an alias to "ls --color".
ls -al |more	List the content of the current directory, all files (also those starting with a dot), and in a long form. Pipe the output through the "more" command, so that the display pauses after each screenful.
cd *directory*	Change directory. Using "cd" without the directory name will take you to your home directory. "cd -" will take you to your previous directory and is a convenient way to toggle between two directories.
cp *source destination*	Copy files.
mcopy *source destination*	Copy a file from/to a DOS filesystem (no mounting necessary). E.g., `mcopy a:\autoexec.bat ~/junk` . See `man mtools` for related commands: mdir, mcd, mren, mmove, mdel, mmd, mrd, mformat ....
mv *source destination*	Move or rename files. The same command is used for moving and renaming files and directories.
ln *source destination*	Create a hard link called *destination* to the file called *source*. The link appears as a copy of the original files, but in reality only one copy of the file is kept, just two (or more) directory entries point to it.. Any changes the file are automatically visible throughout. When one directory entry is removed, the other(s) stay(s) intact. The limitation of the hard links are: the files have to be on the same filesystem, hard links to directories or special files are impossible.
ln -s *source destination*	Create a symbolic (soft) link called "destination" to the file called "source". The symbolic link just specifies a path where to look for the file. In contradistinction to hard links, the source and destination don't not have to tbe on the same filesystem. In comparison the hard links, the drawback of symbolic links are: if the orignal file is removed, the link is "broken", symbolic links can also create circular references (like circular references in spreadsheets or databases, e.g., "a" points to "b" and "b" points back to "a").
rm *files*	Remove (delete) files. You must own the file in order to be able to remove it.
mkdir *directory*	Make a new directory.
rmdir *directory*	Remove an empty directory.
rm -r *files*	(recursive remove) Remove files, directories, and their subdirectories. Careful with this command as root--you can easily remove all files on the system with such a command executed on the top of your directory tree, and there is no undelete in Linux (yet). But if you really wanted to do it (reconsider), here is how (as root): `rm -rf /*`
cat *filename* | more	View the content of a text file called "filename", one page a time. The "|" is the "pipe" symbol (on many American keyboards it shares the key with "\") The pipe makes the output stop after each screenful. For long files, it is sometimes convenient to use the commands `head` and `tail` that display just the beginning and the end of the file. If you happend to use "cat" a binary file and your terminal displays funny characters afterwards, you can restore it with the command "reset".

less *filename*	Scroll through a content of a text file. Press q when done. "Less" is roughly equivalent to "more" , the command you know from DOS, although very often "less" is more convenient than "more".
pico *filename*	Edit a text file using the simple and standard text editor called `pico`.
pico -w *filename*	Edit a text file, while disabling the long line wrap. Handy for editing configuration files, e.g. `/etc/fstab`.
lynx *file.html*	View an html file or browse the net from the text mode.
tar -zxvf *filename.tar.gz*	Untar a tarred and compressed tarball (*.tar.gz or *.tgz) that you downloaded from the Internet.
tar -xvf *filename.tar*	Untar a tarred but uncompressed tarball (*.tar).
gunzip *filename.gz*	Decompress a zipped file (*.gz" or *.z). Use gzip (also `zip` or `compress`) if you wanted to compress files.
bunzip2 *filename.bz2*	Decompress a file (*.bz2) zipped with bzip2 compression utility. Used for big files.
unzip *filename.zip*	Decompress a file (*.zip) zipped with a compression utility compatibile with PKZIP for DOS.
find / -name "*filename*"	Find the file called "filename" on your filesystem starting the search from the root directory "/" . The "filename" may contain wildcards (*,?).
locate *filename*	Find the file name of which contains the string "filename". Easier and faster than the previous command but depends on a database that normally rebuilds at night.
pine	A good text-mode mail reader. Another good and standard one is "elm". Your Netscape mail will read the mail from your Internet account. Pine will let you read the "local" mail, e.g. the mail your son or a cron process sends to you from a computer on your home network. The command `mail` could also be used for reading/composing mail, but it would be inconvenient--it is meant to be used in scripts for automation.
talk *username1*	Talk to another user currently logged on your machine (or use "`talk` *`username1@machinename`*" to talk to a user on a different computer) . To accept the invitation to the conversation, type the command "`talk` *`username2`*". If somebody is trying to talk to you and it disrupts your work, your may use the command "`mesg n`" to refuse accepting messages. You may want to use "`who`" or "`rwho`" to determine the users who are currently logged-in.
mc	Launch the "Midnight Commander" file manager (looks like "Norton Commander" for Linux).
telnet *server*	Connect to another machine using the TELNET protocol. Use a remote machine name or IP address. You will be prompted for your login name and password--you must have an account on the remote machine to login. Telnet will connect you to another machine and let you operate on it as if you were sitting at its keyboard (almost). Telnet is not very secure--everything you type goes in open text, even your password!
rlogin *server*	(=remote login) Connect to another machine. The login name/password from your current session is used; if it fails you are prompted for a password.
rsh *server*	(=remote shell) Yet another way to connect to a remote machine. The login name/password from your current session is used; if it fails you are prompted for a password.

ftp *server*	Ftp another machine. (There is also `ncftp` which adds extra features and `gftp` for GUI .) Ftp is good for copying files to/from a remote machine. Try user "anonymous" if you don't have an account on the remote server. After connection, use "?" to see the list of available ftp commands. The essential ftp command are: `ls` (see the files on the remote system), `ASCII`, `binary` (set the file transfer mode to either text or binary, important that you select the proper one), `get` (copy a file from the remote system to the local system), `mget` (get many files at once), `put` (copy a file from the local system to the remote system), `mput` (put many files at once), `bye` (disconnect). For automation in a script, you may want to use `ncftpput` and `ncftpget`, for example: `ncftpput -u my_user_name -p my_password -a remote.host.domain remote_dir *local.html`
minicom	Minicom program (looks like "Procomm for Linux").
./program_name	Run an executable in the current directory, which is not on your PATH.
xinit	Start a barebone X-windows server (without a windows manager).
startx	Start an X-windows server and the default windows manager. Works like typing "win" under DOS with Win3.1
startx -- :1	Start another X-windows session on the display 1 (the default is opened on display 0). You can have several GUI terminals running concurrently. Switch between them using <Ctrl><Alt><F7>, <Ctrl><Alt><F8>, etc.
xterm	(in X terminal) Run a simple X-windows terminal. Typing `exit` will close it.
xboing	(in X terminal). Very nice, old-fashioned game. Many small games/programs are probably installed on your system....
gimp	(in X terminal) A humble looking but very powerful image editor. Takes some learning to use, but it is great for artists. Use your mouse right button to get local menus.
netscape	(in X terminal) Run netscape (requires a separate Netscape installation). The current versions of Netscape (4.x) are known to be big and buggy. They occasionally crash by vanishing (no other harm done). Also, when not connected to the network , Netscape likes to refuse to do anything (looks like it hanged)-it revives when you connect.
netscape -display *host*:0.0	(in X terminal) Run netscape on the current machine and direct the output to machine named "host" display 0 screen 0. Your current machine must have a permission to display on the machine "host" (typically given by executing the command `xhost` *`current_machine_name`* in the xterminal of the machine host.
shutdown -h now	(as root) Shut down the system to a halt. Mostly used for a remote shutdown. Use <Ctrl><Alt><Del> for a shutdown at the console (which can be done by any user).
halt reboot	(as root, two commands) Halt or reboot the machine. Used for remote shutdown, simpler to type than the previous command.
man *topic*	Display the contents of the system manual pages (help) on the topic. Try "`man man`" first. Press "q" to quit the viewer. The command "`info` *`topic`*" works similar and may contain more up-to-date information. Manual pages can be hard to read. Try "*`any_command`* `--help`" for short, easy to digest help on a command. If more info needed, have a look to the directory /usr/doc
apropos *topic*	Give me the list of the commands that have something to to do with my topic.

6.4 Process control

ps	(=print status) Display the list of currently running processes with their process IDs (PID) numbers. Use "`ps axu`"to see all processes currently running on your system (also those of other users or without a controlling terminal), each with the name of the owner. Use "top" to keep listing the processes currently running.
fg *PID*	Bring a background or stopped process to the foreground.
bg *PID*	Send the process to the background. Opposite to fg. The same can be accomplished with <Ctrl>z.
any_command&	Run any command in the background (the symbol "&" means "run the proceeding command in the background").
kill *PID*	Force a process shutdown. First determine the PID of the process to kill using ps.
killall *program_name*	Kill program(s) by name.
xkill	(in an xwindow terminal) Kill a GUI-based program with mouse. (Point with your mouse cursor at the window of the process you want to kill and click.)
lpc	(as root) Check and control the printer(s). Type "?" to see the list of available commands.
lpq	Show the content of the printer queue. Under KDE (X-Windows), you may also use GUI-based "Printer Queue" available from "K"menu-Utilities.
lprm *job_number*	Remove a printing job "job_number" from the queue.
nice *program_name*	Run *program_name* adjusting its priority. Since the priority is not specified in this example, it will be adjusted by 10 (the process will run slower), from the default value (usually 0). The lower the number (of "niceness" to other users on the system), the higher the priority. The priority value may be in the range -20 to 19. Only root may specify negative values. Use "top" to display the priorities of the running processes.
renice -1 *PID*	(as root) Change the priority of a running process to -1. Normal users can only adjust processes they own, and only up from the current value (make them run slower).

Please note also the control shortcuts: <Ctrl>c, <Ctrl>z, <Ctrl>s, and <Ctrl>q. They were described previously [p 107] .

6.5 Basic administration commands

printtool	(as root in X-terminal) Configuration tool for your printer(s). Settings go to the file /etc/printcap.
setup	(as root) Configure mouse, soundcard, keyboard, X-windows, system services.
linuxconfig	(as root, either in text or graphical mode). You can access / change hundereds of setting from it. Very powerful--don't change too many things at the same time, and be careful with changing entries you don't understand.
xvditune	(in X-terminal). Adjust the settings of the graphical display for all resultions so as to eliminate black bands, shift the display right/left/up/down, etc. (first use the adjustements on your monitor to fit your text mode correctly on the screen). To make the changes permanent, display the frequencies on the screeen and transfer them to the setup file `/etc/X11/XF86Config` .
alias ls="ls --color"	Create an alias for the command "ls" to enhance its format with color. In this example, the alias is also called "ls". Put the alias into the file `/etc/bashrc` if you would like the alias to be always accessible to all users on the system. Type "`alias`" alone to see the list of aliases on your system.
adduser *user_name*	Create a new account (you must be root). E.g., `adduser barbara` Don't forget to set up the password for the new user in the next step. The user home directory is `/home/`*`user_name`*`.`
useradd *user_name*	The same as the command " `adduser` *`user_name`* ".

userdel *user_name*	Remove an account (you must be a root). The user's home directory and the undelivered mail must be dealt with separately.
groupadd *group_name*	Create a new group on your system. Non-essential on a home machine with a small number of users.
passwd	Change the password on your current account. If you are root, you can change the password for any user using: `passwd user_name`

chmod *perm filename*	(=change mode) Change the file access permission for the files you own (unless you are root in which case you can change any file). You can make a file accessible in three modes: read (r), write (w), execute (x) to three classes of users: owner (u), members of the same group as the owner (g), others on the system (o). Check the current access permissions using:
	`ls -l filename`
	If the file is accessible to all users in all modes it will show:
	`rwxrwxrwx`
	The first triplet shows the file permission for the owner of the file, the second for his/her group, the third for others. A "no" permission is shown as "-".
	E.g., this command will **add** the permission to read the file "junk" to all (=user+group+others):
	`chmod a+r junk`
	This command will remove the permission to execute the file junk from others:
	`chmod o-x junk`
	Also try here for more info.
	You can set the default file permissions for the news files that you create using the command "`umask`" (see `man umask`).

chown *new_ownername filename* chgrp *new_groupname filename*	Change the file owner and group. You should use these two commands after you copy a file for use by somebody else.
su	(=substitute user id) Assume the superuser (=root) identity (you will be prompted for the password). Type "exit" to return you to your previous login. Don't habitually work on your machine as root. The root account is for administration and the su command is to ease your access to the administration account when you require it.
rpm -ivh *filename.rpm*	(=RedhatPackageManager, install, verbose, hashes displayed to show progress, as root.) Install a content of RedHat rpm package(s) and print info on what happened. Keep reading if you prefer a GUI installation.
rpm -qpi *filename.rpm*	(=RedhatPackageManager, query, package, list.) Read the info on the content of a yet uninstalled package *filename.rpm*.
rpm -qpl *filename.rpm*	(=RedhatPackageManager, query, package, information.) List the files contained in a yet uninstalled package *filename.rpm*.
rpm -qf *filename*	(=RedhatPackageManager, query, file.) Find out the name of the *.rpm package to which the file *filename* (on your hardrive) belongs.
rpm -e *packagename*	(=RedhatPackageManager, erase=uninstall.) Uninstall a package *pagckagename*. *Packagname* is the same as the begining of the *.rpm package (without the dash and versionnumber).
glint gnorpm kpackage	(in X terminal, as root if you want to be able to install packages) Nice GUI fronts to the Red Hat Package Manager (rpm). "glint" comes with RH5.2, "gnorpm" with RH6.0, "kpackage" must be intalled separately but is the best of the three. Use any of them to to view the software packages installed on your system, the not-installed software packages available on your RedHat CD, display the info about them, and install them if you want (installation must be done as root).
kernelcfg	(as root in X terminal). GUI to to add/remove kernel modules. You can do the same from the command line using the command "`insmod`", but "`insmode`" is less "newbie-friedly".

lsmode	List currently loaded kernel modules.
modprobe -l |more	List all the modules available for your kernel.
insmod parport insmod ppa	(as root) Insert modules into the kernel (a module is roughly an equivalent of a DOS device driver). This example shows how to insert the modules for support of the external parallel port zip drive (it appears to be a problem to get the external zip drive to work in any other way under RH6.0).
rmmod *module_name*	(as root, not essential). Remove the module *module_name* from the kernel.
setserial /dev/cua0 port 0x03f8 irq 4	(as root) Set a serial port to a non-standard setting. The example here shows the standard setting for the first serial port (cua0 or ttyS0). The standard PC settings for the second serial port (cua1or ttyS1) are: address of i/o port 0x02f8, irq 3. The third serial port (cua2 or ttyS2): 0x03e8, irq 4. The forth serial port (cua3 or ttyS3): 0x02e8, irq 3. Add your setting to /etc/rc.d/rc.local if you want it to be set at the boot time. See man setserial for good a overview.
fdisk	(as root) Linux hard drive partitioning utility (DOS has a utility with the same name).

cd
/usr/src/linux-2.0.36
make xconfig
	(as root in X terminal). Nice GUI front-end for configuration of the kernel options in preparation for compilation of your customized kernel. (The directory name contains the version of your Linux kernel so you may need to modify the directory name if your Linux kernel version is different than 2.0.36 used in this example. You also need the "Tk" interpreter and the kernel source code installed.) The alternatives to "make xconfig" are: "make config" (runs a scripts that asks you questions in the text mode) and "make menuconfig" (runs a text-based menu-driven configuration utility). Try: `less /usr/doc/HOWTO/Kernel-HOWTO` for more information.
	After the configuration, you may choose to proceed with kernel compilation of the new kernel by issuing the following commands:
	`make dep`
	`make zImage`
	The last command will take some time to complete (maybe 0.5 h, depending on your hardware). It produces the file "zImage", which is your new Linux kernel. Next:
	`make modules`
	`make modules_install`
	Read: /usr/doc/HOWTO/Kernel-HOWTO for information on how to install the new kernel. You will probably also find it useful to read "man depmode". Configuration, compilation and installation of a new kernel is not difficult but it CAN lead to problems if you don't know what you are doing. Compilation of a kernel is a good way to test your hardware, because it involves a massive amount of computing. If your hardware is "flaky", you will most likely receive the "signal 11" error (read the beatiful `/usr/doc/FAQ/txt/GCC-SIG11-FAQ`). **See this for details on kernel upgrade.**

depmod -a	(as root) Build the module dependency table for the kernel. This can, for example, be useful after installing and booting a new kernel. Use "`modprobe -a`" to load the modules.
ldconfig	(as root) Re-create the bindings and the cache for the loader of dynamic libraries ("ld"). You may want to run ldconfig after an installation of new dynamically linked libraries on your system. (It is also re-run every time you boot the computer, so if you reboot you don't have to run it manually.)
touch *filename*	Change the date/time stamp of the file *filename* to the current time. Create an empty file if the file does not exist.
mknod /dev/fd0 b 2 0	(=make node, as root) Create a device file. This example shows how to create a device file associated with your first floppy drive and could be useful if you happened to accidentlally erase it. The options are: b=block mode device (c=character mode device, p=FIFO device, u=unbuffered character mode device). The two integers specify the major and the minor device number.
fdformat /dev/fd0H1440 mkfs -c -t ext2	(=floppy disk format, two commands, as root) Perform a low-level fromatting of a floppy in the first floppy drive (/dev/fd0), high density (1440 kB). Then make a Linux filesystem (-t ext2), checking/marking bad blocks (-c). Making the files system is an equivalent to the high-level format.
badblocks /dev/fd01440 1440	(as root) Check a high-density floppy for bad blocks and display the results on the screeen. The parameter "1440" specifies that 1440 blocks are to be checked. This command does not modify the floppy.

fsck -t ext2 /dev/hda2	(=file system check, as root) Check and repair a filesystem. The example uses the partition hda2, filesystem type ext2.
dd if=/dev/fd0H1440 of=floppy_image dd if=floppy_image of=/dev/fd0H1440	(two commands, dd="data duplicator") Create an image of a floppy to the file called "floppy_image" in the current directory. Then copy `floppy_image` (file) to another floppy disk. Works like DOS "DISKCOPY".

See here for details on mounting. This are summarizing examples:

mount -t auto /dev/fd0 /mnt/floppy	(as root) Mount the floppy. The directory `/mnt/floppy` must exist, be empty and NOT be your current directory.
mount -t auto /dev/cdrom /mnt/cdrom	(as root) Mount the CD. You may need to create/modify the `/dev/cdrom` file depending where your CDROM is. The directory `/mnt/cdrom` must exist, be empty and NOT be your current directory.
mount /mnt/floppy	(as user or root) Mount a floppy as user. The file `/etc/fstab` must be set up to do this. The directory `/mnt/floppy` must not be your current directory.
mount /mnt/cdrom	(as user or root) Mount a CD as user. The file `/etc/fstab` must be set up to do this. The directory `/mnt/cdrom` must not be your current directory.
umount /mnt/floppy	Unmount the floppy. The directory `/mnt/floppy` must not be your (or anybody else's) current working directory. Depending on your setup, you might not be able to unmount a drive that you didn't mount.

6.6 Networking tools

netconf	(as root) A very good menu-driven setup of your network.
ping *machine_name*	Check if you can contact another machine (give the machine's name or IP), press <Ctrl>C when done.
route -n	Show the routing table.
nslookup *host_to_find*	Query your default domain name server (DNS) for an Internet name *host_to_find*.
ipfwadm -F -p m	(only RH5.2 or lower) Set up the firewall IP forwarding policy to masquerading. (Not very secure but simple.) Purpose: all computers from your home network will appear to the outside world as one very busy machine and, for example, you will be allowed to browse the Internet from all computers at once.
ipfwadm-wrapper -F -p deny ipfwadm-wrapper -F -a m -S xxx.xxx.xxx.0/24 -D 0.0.0.0/0	(two commands, RH6.0). Does the same as the previous command. Substitute the "x"s with digits of your class "C" IP address that you assigned to your home network. See here for more details.
ifconfig	(as root) Display info on the network interfaces currently active (ethernet, ppp, etc). Your first ethernet should show up as eth0, second as eth1, etc, first ppp over modem as ppp0, second as ppp1, etc. The "lo" is the "loopback only" interface which should be always active. Use the options (see `ifconfig --help`) to configure the interfaces.

ifup *interface_name*	(`/sbin/ifup` to it run as a user) Startup a network interface. E.g.:
	`ifup eth0`
	`ifup ppp0`
	Users can start up or shutdown the ppp interface only when the right permission was checked during the ppp setup (using `netconf`).

ifdown *interface_name*	(`/sbin/ifdown` to run it as a user). Shut down the network interface. E.g.: `ifdown ppp0` Also, see the previous command.
netstat | more	Displays a lot (too much?) information on the status of your network.

6.7 Linux built-in developer tools and programming languages

emacs	(in X-terminal) The emacs editor. For gurus only. Yes, the "vi" editor is also available (not recommended for newbies). To exit "vi" press: <Esc> : q!<Enter>
gcc *filename.c*	GNU C compiler. Quite straight-forward. Good free manuals are available on the net.
g++ *filename.C*	GNU C++ compiler. The capital "C" is normally used for C++ sources.
perl	Powerful and widely used scripting language. Flexible but with cryptic syntax. Very popular among gurus.
python	Modern and elegant object oriented interpreter. Seems as powerful and easier than perl. Very good (and large) free handbooks by G. van Rossum are available on the net (try: http://www.python.org/doc/ for browsing or ftp://ftp.python.org for downloading).
tcl	(Often pronounced "tickle".) Popular scripting language.
wish	(xterm) A front end to Tk, an X-windows extension of tcl. Often used for building front-ends of a program.
guile	An implementation of "Scheme" programming language.
g77	GNU FORTRAN. If you are into FORTRAN, you may want tocheck: http://studbolt.physast.uga.edu/templon/fortran.htmlto find a FORTRAN compiler that suits your particular needs under Linux.
f2c	FORTRAN to c converter.
fort77	FORTRAN compiler. Invokes f2c and then gcc or g++.
basic	"Chipmunk Basic".
make	Run the "make" utility to compile a project described in the Makefile found in the current directory.
cvs	Concurrent versions system. Try: info cvs
diff *file1 file2* > *patchfile*	Compare content of two files and list any differences. Save the output to the file *patchfile*.
sdiff *file1 file2*	Side-by-side comparison of two text files.
patch *file_to_patch patchfile*	Apply the patch (a file produced by diff, which lists differences between two files) *patchfile* to the file *file_to_patch.*
grep	Search content of text files for matching patterns. Definitely worth to learn at least the basics of this command. The patterns are specified using a powerful and standard notation called "regular expressions".
tr	Translation utility (e.g. replace characters in a text file).
gawk	GNU awk (mostly for processing delimited text files).
sed	Tool for processing text files.
strings *filename* | more	Display the strings contained in the binary file called filename.

Go to Part 7: Essential Linux applications (proprietary or not)

LINUX NEWBIE ADMINISTRATOR GUIDE

ver. 0.59 1999-11-07

Part 7: Some Essential Linux Applications (proprietary or not)

by Peter, Stan, and Marie Klimas

Answers to Some Frequently Asked Linux Questions

Distributed under the General Public Licence http://www.gnu.org/copyleft/gpl.html. Your feedback, comments, corrections, and improvements are appreciated. Send them to bklimas@magma.ca

---

Quick site navigation:

Start: Linux Newbie Administrator Guide
Part 0: For the Undecided (Linux Benefits)
Part 1: Before Linux Installation
Part 2: Linux Resources, Help and Some Links
Part 3: Basic Operations FAQ
Part 4: Linux Newbie Administrator FAQ
Part 5: How to Upgrade the Kernel
Part 6: Linux Shortcuts and Commands
Part 7: >Essential Linux applications (proprietary or not)<

Contents of this page:

7. Some essential Linux applicatons (propietary or not):

7.1 Word Perfect 8 for Linux [p 120]
7.2 Star Office Suite [p 120]
7.3 Applixware [p 121]
7.4 AbiWord [p 121]
7.5 Spreadsheet [p 121]
7.6 Databases [p 122]
7.7 CAD [p 122]
7.8 Netscape and Lynx (web browsers) [p 123]
7.9 Klyx and Latex [p 124]
7.10 Writing CD-Rs (cdrecord and cdparanoia) [p 126]

---

7. Some essential Linux applications (proprietary or not):

This is not a complete coverage of Linux applications. Selection criteria were:
o The application is not installed during the default RedHat installation--the user must take active steps to find / install it.
o We particularily like this application, or a reputable source led us to believe that it is a really worth

it recommendation.

There are thousands of Linux programs. If you are unsatisfied with our lean choice, try: http://www.linuxapps.com/ or http://stommel.tamu.edu/~baum/linuxlist/linuxlist/linuxlist.html or http://www.boutell.com/lsm/ or http://www.linuxlinks.com/Software/

7.1 Word Perfect 8 for Linux

This is a very good and powerful word processor. A free version is available for personal use. Try: http://linux.corel.com/linux8/download.htm (free registration of the program is required). The free version lacks the equation editor, built-in graphics editor, chart editor and the "art text", but it is otherwise a fully functional version of the best word processor in the world. The full version costs about $50.

File-level compatibility with MS Word is very good--I encountered cases that suggest WP8 may actually be more compatibile with the different "sub-versions" of MS Word fileformat than MS Word itself.

On the down side, Word Perfect for Linux often feels slow, particularly when scrolling larger documents.

WP8 is normally launched through an icon or from the K-menu but you have to add this yourself after the installation. The main executabe is `/usr/local/wp8/wpbin/xwp`, but the location may vary, depending where you installed it. Use the command

```
locate xwp
```

or

```
find / -name "xwp"
```

if you cannot find it.

To "manually" launch WP8, try in X-terminal:

```
/usr/local/wp8/wpbin/xwp
```

Corel plans to lunch its own Linux distribution in late 1999/early 200 (based on Debian) , which apparently will center around their powerful WordPerfect suite (WordPerfect wordprocessor, Quattro Pro spreadsheet, etc) different from their current offerring on Linux in that it will be based on the GPL "Wine" (=MS Windows-Aplication Programming Interface-Emulation) library.

7.2 Star Office Suite

Star Office is a complete office suite: word processor, spreadsheet and presentation program. Full version is free for personal use--it can be downloaded over the Internet: try http://www.stardivision.com/freeoffice/ (large, 60-80 MB download, probably not practical with a

modem).

Star Office looks and acts very much like MS Office for Windows (this includes that it is big and slow) and may not be worth the trouble without at least 32 MB of physical memory. It also has a good file-level compatibility with MS Office (read and write MS Word, MS Excel and MS PowerPoint file formats).

Older versions of Star Office required registration after which you obtain a "registration key". The registration key is based on your e-mail which you give to Star Office Corp., so make sure you remember which e-mail you gave them and enter the same to the program when it pops the registration box, together with the key. The latest versions (after StarOffice was purchased by Sun) does not anything like that any more.

7.3 Applixware

Applixware is another complete office suite. It contains a word processor, spreadsheet, graphics, presentation, mail, html authoring, and a few more applications. Many say it's the best. I don't know because no free edition is available. See http://www.applix.com/appware/linux/index.htm for more information. If you are really desparate for Aplixware, you can download the demo at: http://www.download.com/, but do you really want to? [Sorry, I have to say this: why do copyright lawers keep killing perfectly good software?]

7.4 AbiWord

AbiWord (http://www.abisource.com) is a good light-weight wordprocessor. Worth trying for simple word processing needs. Although still fairly incomplete, it is quite useful to me, e.g. it supports spelling-as-you-type without the overhead of WordPerfect.

7.5 Spreadsheet

Besides the spreadsheet applications in the suites above (StarOffice and Applixware), you may want to have a look at "xess": http://www.ais.com/linux_corner.html .

They have also a slightly scaled down shareware "xesslite" version which can be downloaded from their site. We love the general design of xess (simple and almost self-explanatory). It seems to have really good power for engineering applications and in our opinion it may be the best spreadsheet currently available on Linux.

You may want to keep your eye at "Gnumeric" (included with your RH6.0 distribution, part of the Gnome project). It is still relatively incomplete but it is already definitely usable. It is under heavy development and definitely has the potential to become really great in the near future--it already has a lot of built-in functions, but its printing is unreliable. You can start it up (if you installed it) by typing in the X-windows terminal:

`gnumeric`

Kspread is another highly promising spreadsheet. It is part of the KDE project to be integrated with the KDE2.0, most likely you don't have it on your system and I would not advice you to try to install it--it requires next-generation KDE libraries and you can screw up your KDE if you try to install these.

Both gnumeric and kspread file format is xml (the already standard, next-generation, enhanced html). This file format is definitely good news if you ever experienced problems with MS-Windows-based spreadsheet file formats.

7.6 Databases

If you are a database person, you will be pleased to see that Linux is very well covered in this area.

postgreSQL is a high-powered database available on your RH5.2 CD (free, unrestrictive licence).

mySQL http://www.MySQL.com/ is free for non-commercial use (and also free for commercial use under Linux, I am told).

Sybase for Linux (edited for space):

> From: Rumy Driver <rdriver@sybase.com> Organization: Sybase Inc.
> [...] I do work for Sybase and we have a full-featured Y2K database for GNU/Linux which is free of cost for unlimited use. Available on the RedHat 5.2 distribution (on the 3rd CD). Rumy Driver

Check: http://www.sybase.com:80/sqlserver/linux/aselinux_install.html for more information.

Interbase for Linux can be downloaded free for personal use:
http://www.interbase.com/downloads/products.html .

There is also **Oracle** for Linux: http://platforms.oracle.com/linux/index_lin.htm. For an Oracle-Linux howto, see: http://jordan.fortwayne.com/oracle/index.html

7.7 CAD

QCAD (GPL): http://www.qcad.org
OCTREE (free for non-commercial): http://www.octree.de/html/frames/eng/f_octree.htm
VariCAD (proprietary commercial): http://www.varicad.com/
VARKON (LGPL): http://www.varkon.com/
Microstation (prorpietary): http://www.microstation.com/academic/products/linux.htm--the academic edition of Microstation includes the Linux version of their excellent CAD system (better than AutoCad).

7.8 Netscape and Lynx (web browsers)

You may have installed a 4.xx version of Netscape during your RedHat installation. To run it, try (in X-terminal):

```
netscape
```

If you didn't install Netscape, you may want to put your RedHat CD into the CDROM, mount it using, for example (as root):

```
mount -t auto /dev/cdrom /mnt/cdrom
```

and then start, from X-terminal: `glint` (for RH5.2)or `gnorpm`(for RH6.0) to browse the available packages and perhaps install netscape from your RedHat CD. If you prefer to do the installation from from the command line, try, after mounting the CDROM:

```
cd /mnt/cdrom
cd RedHat/RPMS
rpm -i netsca*
```

Netscape is a very good browser, with the same look, feel, and power as Netscape for MS Windows, so you will have no problems navigating it. On the dark side, Netscape sometimes crashes (just disappears from the screen, no damage done, you have to restart it). Also, in some configurations, Netscape does not like to be be run without a connection to the Internet (depending on your configuration, Netscape can take it up to a couple of minutes to figure out that there is no connection). Any Netscape version 4.xx seems to have similar problems, so upgrading to a slightly higher version does not seem to help. Take your heart, once tuned-up, Netscape runs quite well, and is quite definitely perfectly usable (I use it all the time). Gecko (the already famous, revolutionary offspring of Netscape) is under way so hold your breath :-) .

For now, if your Netscape is too buggy to you, you may want to empty your mail "trash" bin and compress the mail folders on regular basis, and disable Java (in the menu "Edit-Preferences-Advanced"). This seems to make Netscape more stable. If Netscape "freezes" when you work off-line, disabling the automatic checkup for e-mail solved the problem for me.

If your Netscape crashes and on a subsequent re-start complains about the presence of a lock file, it might have left a lock file in your home directory. Just quit any instance of Netscape that you might be running and delete the "lock" file from the directory ".netscape" under your home directory:

```
cd ~
cd .netscape
rm lock
```

If your ISP connection is really slow, you may prefer a text-based browser:

```
lynx
```

which is a real piece of art and does not have any problems whatsoever.

Another choice of a Internet browser is your KDE help utility (click on the "book with the lightbulb" button on you K-bar). This is really an html browser so if you are connected to the Internet (e.g. using your kppp), you can browse almost anything using this simple utility.

To compose html pages, I use Netscape (WYSiWYG view) and WebMaker (code view).

7.9 Klyx and Latex

Klyx is a "K-desktop" variant of Lyx. Both Klyx and Lyx are front ends (WYSIWYG, running under X-Windows) of Latex. Latex has for years been the most popular document prepartion and typsetting program, particularly in academia (good with equations, etc.).

The good news is that even if you do not know what Latex is, you may still be able to use Klyx. Think of Klyx as a word processor, although its philosophy is different from that of other word processors these days, and therefore it may require an adjustment of your mindset. Latex (and Klyx) philosophy is to type in the text, define the "styles" and leave the formatting to the typesetting program. This means you never adjust the spacing (between words, sentences, paragraphs, chapter, etc.) manually. When done with your document, you "compile" your text to create a device independent file ("*.dvi"). The *.dvi file can be viewed using a dvi viewer and printed. The quality of the output is usually outstanding, but its creation process is typically somewhat more frustrating than using a regular word processor.

The strength of Latex is the quality of the printouts, its capability to cope with long, complex documents (technical books, etc.), availability of all foreign characters, and its portability across many different platforms.

Try it out, Klyx is free. As almost any piece of Linux software, you can download it from Linuxberg:

http://idirect.linuxberg.com/kdehtml/off_word.html

or any other fine Linux software depository on the Net.

If instead of easier Klyx, you wanted to try straight, hard-core Latex, here is some info to get you started:
* Use your favourite text editor to create a Latex document, spellcheck it, etc., save the text file with the extension "*.tex". Read on to see my sample Latex document.
* Envoke Latex to "compile" the text file into a "*.dvi" ("device indepenent") file by typing on the command line:

```
latex my_letex_file.tex
```

* Print the "my_latex_file.dvi" file which was created by the previous command by invoking the dvi to postscript utility, that on default send the output to the lpr printer:

```
dvips my_file.dvi
```

You can also save the output to postscript file by typing:

```
dvips -o output_file.ps my_file.dvi
```

Here is my sample Latex file:

```
% Any line starting with "%" is a comment.
% "\" (backslash) is a special Latex character which introduces a Latex
% command.

\documentclass[10pt]{article}
\begin{document}

% Three commands are present in every Latex document. Two of them are
% above and one at the very end of this sample document.

This is a simple document to try \LaTeX. Use your favourite plain text
editor to type in your text. See how the command \LaTeX produces the
\LaTeX logo. Here is the end of the first paragraph.

Here starts the second paragraph (use one or more empty lines in your
input file to introduce a new paragraph).

The document class of this sample is ‘‘article’’ and it is defined at the
very beginning of the document. Other popular classes are ‘‘report’’,
‘‘book’’ and ‘‘letter’’.

Please note that the double quote is hardly ever used, utilize
two ‘ to begin a quote and two ’ to close it.

Here are different typefaces:
{\rm This is also roman typeface. It is the default typeface.}
{\bf This is bold typeface. }
{\em This is emphesize (italic) typeface.}
{\sl This is slanted typeface, which is different from the italic.}
{\tt This is typewriter typeface.}
{\sf This is sans serif typeface.}
{\sc This is small caps style.}

You can itemize things:
\begin{itemize}
\item one
\item two
\item three
\end{itemize}

You can also enumerate things:
\begin{enumerate}
\item one
\item two
\item three
\end{enumerate}
```

```
Try some foreign letters and symbols:
\aa \AA \o \O \l \L \ss \ae \AE \oe \OE \pounds \copyright \dag \ddag \S
\P. There are also three dashes of different length: - -- ---.

Try some accents over the letter ‘‘a’’: \’{a} \‘{a} \"{a} \^{a} \~{a}
\={a} \.{a} \b{a} \c{a} \d{a} \H{a} \t{a} \u{a} \v{a}. Other letters can
be accented in a similar way.

The pair of ‘‘\$’’ marks a math context. Many special characters are
available only in the ‘‘math’’ context. For example, try the Greek
alphabet:

Small: $ \alpha \beta \gamma \delta \epsilon \varepsilon \zeta \eta
\theta \vartheta \iota \kappa \lambda \mu \nu \xi o \pi \varpi
\rho \varrho \sigma \varsigma \tau \upsilon \phi \varphi
\chi \psi \omega $

Capital: $ A B \Gamma \Delta E Z H \Theta I K \Lambda M \Xi \Pi P
\Sigma T \Upsilon \Phi X \Psi \Omega $

Try some equations: $ x^{y+1} + \sqrt{p \times q}=z_{try_subscripts} $

\begin{center}
$ \frac{x \times y}{x/2+1}=\frac{1}{3} $
\end{center}

\LaTeX math commands are very similar to those in the old ‘‘Word Perfect’’
equation editor.

Use the verbatim mode to print the 10 special symbols which normally have
special meaning in \LaTeX: \verb|%${}_#&^~\|. The special symbols must be
contained between any two identical characters which in the example above
is |. Most of these special symbols can also be printed by preceeding the
character with a backslash: \% \$ \{ \} \_ \# \& \^.

% This command ends the document (this is the third one that *must* be
% present in every document).
\end{document}
```

7.10 Writing CD-Rs (cdrecord and cdparanoia)

Disclaimer: Copying copyrighted material is illegal. Do NOT use the instructions below for anything illegal.

INTRO

Setting up a CD writer can be tricky. For good info see the page http://www.guug.de/~winni/linux/cdr/html/CD-Writing-3.html .

I used the above document to set up a cheap, no-name IDE-ATAPI CD writer on a PC running RH6.0. It works great, and no changes to my setup were required after the recent upgrade to RH6.1. Here are the steps I followed (almost everyghing has to be done as root):

SETUP

o In the file `/etc/lilo.conf` , add a line at the end of the Linux "image" section:

```
append="hdb=ide-scsi"
```

Adjust the line above if your CD writer is not "hdb" (second drive on the first IDE interface). It makes your IDE-ATAPI CD-W(R) to be seen on your Linux system as a SCSI device. (It is not really a SCSI device, it is an IDE device, it just pretends to be SCSI.) Run `lilo` after making any changes to `/etc/lilo.conf` . The above change to `/etc/lilo.conf` appears to be necessary to be able to emulate SCSI on IDE-ATAPI CD-R(W) if you use Linux kernel 2.2.x.

o Add the loop devices to the `/dev/` directory. This is not a obligatory, but a nice feature if you plan creating your own data CDs. The loop device will let you mount a CD image file (as if it was a already a filesystem) to inspect its content. The loop devices don't exist on my hard drive after Linux RedHat installation, so I create them using:

```
cd /dev/
./MAKEDEV loop
```

o Add these two lines at the end of the file `/etc/rc.d/rc.local` so that the needed kernel modules are automaticly loaded on system startup:

```
/sbin/insmod ide-scsi
/sbin/insmod loop
```

These two kernel modeules are needed for SCSI emulation of IDE drives and to support the loop devices, respectively.

o Create or modify the device `/dev/cdrom` so it now points to the correct device , most likely:

```
ln -s /dev/sr0 /dev/cdrom
```

You need to do this because "/dev/cdrom" pointed to an IDE device (probably `/dev/hdb`) but now this changes since your CD-R is going to be in SCSI emulation mode.

o Reboot so the changes to `/etc/lilo.conf` can take effect. Check if your CD-R(W) still works properly for normal reading.

o Download the program "cdrecord" from your favorite Linux software repository (e.g., http://idirect.linuxberg.com/). Then install the source code, compile it, install the program, and make symbolic links so that the executable are easy to run (the installation would be much easier if you found a binary *.rpm file):

```
cd /usr/local
tar -xvzf /the_path_to_which_you_downloaded/cdrecord-1.6.1.tar.gz
ls
cd cdrecord-1.6.1
make
```

```
make install
ls /opt/schily/bin/
ln -s /opt/schily/bin/* /usr/local/
```

The program `cdrecord` is a spartan, command line utility for writing CD. There are several GUI front ends to it, but they will be useless if the underlying cdrecord does not work proprely.

o See if your cdwriter is recognized. If it is, it should now show in the output from this command:

```
cdrecord -scanbus
```

DATA CDs

o Create a CD image containing your data:

```
mkisofs -r -o cd_image input_data_directory
```

This makes an International Standard Organization (ISO) standard 9660-type filesystem containing the files from `input_data_directory`, but writes the filesystem to an ordianry file on the hard drive. This output file is an "image" of the new CD which I am creating. The option "-r" sets the file permissions so all the files on the CD are publicly readable (can be read by all user, not only the file owner). The option "-o" indicates that the parameter that follows is the output filename (the "CD image"). The filenames are abbreviated to the "8.3" DOS-type length but, since Linux supports so called "Rock Ridge" extensions to ISO9660, it also writes the full names and all the file permissions as well--this way the new filesystem is portable across all popular operating systems (DOS, MS Windows, Linux, UNIX, etc)--really convenient to the user.

o You may want to inspect the CD image file by mounting it through the loop device:

```
mount -t iso9660 /dev/loop0 cd_image /mnt/cdrom
[now the content of the file should appear in /mnt/cdrom]
cd /mnt/cdrom
[inspect the file mounted through the loop device]
```

When done with inspection, change your working directory away from the mountpoint and unmount the file:

```
cd
umount /mnt/cdrom
```

o If everything worked, you may burn your data CD:

```
cdrecord -v speed=2 dev=1,0,0 -data cd_image
```

The numbers in "dev=" stand for the scsi bus number (the first one is 0, second bus is 1, ...), device id on the scsi bus (between 0 and 7), and the scsi lun numer (always 0) respectively. You must customize them: the first two number can be read in the output from `cdrecord -scanbus`, the third number is 0. Make sure to use the correct numbers or you may write to a wrong drive and corrupt your data.

The timing of writing to CD-Rs is very important, or an error may occur (the laser cannot be switched on and off at will). Therefore avoid doing intensive tasks during creating a CD, e.g. don't create or erase large files on the hard drive. My system will not permit me to start new tasks when using `cdrecord` .

AUDIO CDs

o Audio tracks have to be in files of *.cdr (I guess it is the same as *.cdda.raw), *.wav (wave), or *.au format before you can write them to a CD.

o The utility `sox` converts between the various audio file formats (`sox` understands quite a few of them). For example, this will convert a .wav file to a .cdr file:

```
sox my_file.wav my_file.cdr
```

You don't need to do the conversions manually - `cdrecord` supports *.wav and *.au directly (it does a conversion from *.wav or *.au to *.cdr "on the fly"). This is very convenient because audio files tend to be large.

o Audio CDs don't contain a filesystem, they store "raw data". This means that you cannot mount an audio CD. Also, each track is written separately, i.e., as if it was a different "partition" on the CD.

o To read audio tracks from an audio CD and write them to a suitable file on your hard drive (typical format is *.raw or *.wav) , you need a "cd ripper". A popular CD ripper is "cdparanoia". After downolading the source for cdparanoia, install it (use the autocompletion <Tab> shortcut when typing the long filenames):

```
cd /usr/local
tar -xvzf
/the_path_to_which_you_downloaded/cdparanoia-III-alpha9.6.src.tgz
ls
cd cdparanoia-III-alpha9.6
./configure
make
make install
```

The installation program places a proper link to the executable (`/usr/local/bin/cdparanoia`) so I don't have to create the link manually.

To rip the first track from an audio CD, I can use:

```
cdparanoia 1
```

which will put the first track from the CD into the wave file "cdda.wav" in the current directory.

To rip tracks 1 to 2 from an audio CD to a "raw" file format, I can use:

```
cdparanoia -B -p "1-2"
```

The option -B specifies to use a "batch" mode, so that each track is put into a separate file (this is probably what you want, otherwise all tracks would be placed in one otput file). The "-p" option specifies output in raw format. The files are named `track1.cdda.raw` and `track2.cdda.raw` .

To rip all tracks from an audio CD, each track to a separate *.wav file, while forcing reading speed 4x, I can use:

```
cdparanoia -S 4 -B "1-"
```

Make sure you have sufficient free space on your hard drive. You can use use the space on your DOS partition (if you have dual boot).

o To write suitable audio files to a CD-R(W), I can use:

```
cdrecord -v speed=2 dev=1,0,0 -audio track*
```

o Older stereos often will not play many CD-Rs (this is connected to the size of the pits on the CDs). My home stereo cannot read re-writeable CDs (CD-RW) at all, although it will read write-once disks (CD-Rs), so re-writables may be good to store data but are useless for audio (unless I plan to play them exclusively on my computer).

MIXED-MODE CDs

o Mixed-mode CDs (it is CDs that contain both data and audio, often game CDs) are not a problems, e.g.:

`mount -t iso9660 /dev/cdrom /mnt/cdrom` (mount a mixed-mode CD)
`mkisofs -r -o cd_image /mnt/cdrom` (make an ISO filesystem from the data on the CD).
`umount /mnt/cdrom` (unmount the CD)
`cdparanoia -S 4 -B "2-"` (rip the content of all audio tracks on the CD)
`cdrecord -v speed=2 dev=1,0,0 -data cd_image -audio track*` (write the data and audio files)

OTHER ISSUES

o Re-writable CDs (CD-RW) are used the same way as regular write-once CDs (CD-R), but you have to blank re-writable disks before you will be able to re-use them, e.g.:

```
cdrecord -v speed=2 dev=1,0,0 blank=fast
```

To see other (more thorough and slower) options for blanking, use:

```
cdrecord blank=help
```

o To simplify writing long commands required by `cdrecord`, I may define a global alias by placing the following line in the file `/etc/bashrc`:

```
alias cdrecord="cdrecord -v speed=2 dev=1,0,0"
```

Re-login for the changes in `/etc/bashrc` to take effect. After creating this alias, a I can record a CD using the following shortened command (no need to specify the CD writer speed and device name all the time):

```
cdrecord -audio track*
```

o Most CDs can be copied by first copying all data (for data CDs) or all tracks (for audio CDs) onto the hard drive as described before, but some CDs cannot. For troublesome data CDs, you might try something like:

```
dd conv=noerror,notrunc if=/dev/cdrom of=cd_image
cdrecord -t audio cd_image
```

The `dd` command copies the input file (if), which in this case is the device `/dev/cdrom` to the output file (of) which in this example is a file called `cd_image` (on the hard drive in the current working directory). The option "conv=noerror,notrunc" specifies that the potential reading errors are to be ignored, and files not trancated on an error. The second command copies the file `cd_image` that was created by the `dd` command onto an empty CD.

Return to homepage: Linux Newbie Administrator Guide

Acknowledgments

The Linux Newbie Administrator Guide (LNAG) is hosted FREE OF CHARGE on the SunSite server at Aalborg University, Denmark (http://sunsite.auc.dk/linux-newbie). Thanks to Esben Haabendal Soerensen <bart@sunsite.auc.dk>.

Major help and advice was received from (alphabetical order):

Alan W. Irwin <irwin@beluga.phys.uvic.ca>, Benjamin Smith <bens@saber.net>, Ben McCosh <ben@albagroup.com>, Bill Staehle <staehle@netvalue.net>, Bill Unruh <unruh@physics.ubc.ca>, Brian Kelsay <ripcrd6@kcinter.net>, Ding-Hou Lee <dl105@columbia.edu>, Gary <swear@aa.net>, Greg Mizell <GMizell@peek-traffic.com>, Jeff Greenlee <greenj@analogy.com>, jeff covey <jeff.covey@pobox.com>, Juhapekka Tolvanen <juhtolv@st.jyu.fi>, and Ken Foskey <waratah@zip.com.au>.

Special thanks to the Portland Linux User Group!

Thanks to the several dozen others who sent comments or good word to us. We always appreciate any feedback. We received no flames so far :-) . Thanks to the few thousands who developed GNU/Linux.

Other matters

You can see that this guide was written by newbies for newbies. It should never be considered an authoritative source on any subject--there are much more exhaustive docs, typically more difficult to read too :-), most of them available right on your Linux system in the directory `/usr/doc`. Please e-mail us immediately if you spot a mistake that can confuse or mislead a new Linux user--this work is in progress and the current version may contain such mistakes. Don’t use this guide if your life was to depend on it!

The guide is GPLed (i.e., published under the General Public Licence). If this is not clear, you can link to it, mirror it, copy it, quote it briefly or extensively, print it, translate it, send it to your friend or customer, distribute it, etc. , without our permission. Just be fair, preserve the original copyright notice and keep it free (i.e., GPLed). We will be only glad if you let us know how you used our guide.

Hope this helps. Best regards, the Klimases.

Our rewards and testimonials

The Linux Newbie Administrator Guide (LNAG) has been on-line since February, 1999. Our web pages were visited by perhaps (estimated) 0.25 million readers, and linked to more than 2000 www sites (most perishable though). We are proud that our Guide was:

o Awarded "Link van de dag" ("Link of the Day") by NL.linux.org in March 1999: "Alles wat je als newbie zou moeten weten." http://www.nl.linux.org/index/link_van_de_dag.html

o Mentioned on Linux Today, Apr 14th 1999: http://linuxtoday.com/stories/4985.html

o Mentioned on Linux Weekly News, Apr 22nd 1999: http://lwn.net/1999/0422/announce.phtml

o Mentioned by LinuxWord Czech digest, May, 1999:
http://online.idg.cz:8080/cw_news.nsf/4f682153745289cdc12566b80055b9df/999e7c8725ec0848c1256767004ed0dc
Pøíruèka administrátora pro zaèínající u¾ivatele. Pro zaèínající u¾ivatele Linuxu pøibyl velice zajímavý dokument. Jedná se o pøíruèku "Linux Newbie Administrator", která je psána linuxovým zaèáteèníkem. Autor sám pí¹e, ¾e Linux pou¾ívá pouze devìt mìsícù, ale to je mo¾ná v tomto pøípadì výhoda - je jistì blí¾e potenciálnímu ètenáøi ne¾ nìkterý linuxový guru. Pøíruèka se nachází na webové serveru (http://sunsite.auc.dk/linux-newbie/) a obsahuje pomìrnì jednoduché návody na èinnosti, které jako správce va¹eho linuxového systému budete nucen dìlat. Související URL: sunsite.auc.dk/linux-newbie/

o Awarded "Best of Linux Winner" by Dave Central (subsidiary of Andover), August 20, 1999:
http://linux.davecentral.com/bol_19990820.html
"Best of Linux Archive Getting Linux. Andover. Best of Linux Winner August 20, 1999. Newbie Administrator Guide Tutorials - Getting Started.

"There are two ways to approach Linux use. One is strictly as a user; you can work away, and leave it to a guru friend to fix problems and add new features to your system.

"The do-it-yourself types in the crowd may be satisfied with this for awhile, but eventually, they'll want to get under the hood themselves. The Newbie Administrator Guide is geared toward such folks, and provides a gentle introduction to the craft of maintaining a Linux system. There are other comprehensive works on the subject, such as The Linux System Administrators' Guide, but they're mostly geared toward professional sysadmins who deal with lots of machines with lots of users.

"The Newbie Admin Guide (NAG hereafter) is of more immediate use to the home user who has decided to take on the task of caring and nurturing for her machine herself. Like a lot of great tutorials, the NAG started with users, the Klimas family in this case, who went through the struggle to
set things up and said, "Why should other people go through the same work when we've already done it?" So, they started making notes about what they'd done, and put them on the Web for everyone's benefit.

"The NAG assumes you're starting from ground zero, offering explanations of what Linux is, why you'd want to use it, and where to get it, but moves on to clear and detailed explanations of complex issues you'll face when you try to configure and troubleshoot your system. If you've flipped through book after book but found yourself out of your depth and longing for simple, direct explanations, give the NAG a try.
"

o Mentioned by Linux Today, Oct 25, 1999: http://linuxtoday.com/stories/11525.html

o Recommended by http://www.linuxlinks.com/Beginners/: "Linux Newbie Administrator Guide. Simple answers to problems that Linux newbies frequently encounter while setting up/using/administrating their computer or home network."

o Recommended by http://www.linuxstart.de/einsteiger/index.htm: "Linux Newbie Administrator Guide: Von der Installation, Wartung, Tastaturbefehle bis zur Kernel-Rekompilierung."

o Recommended by http://www.sslug.dk/emailarkiv/novice/1999_04/msg00013.html To: sslug-novice@sslug.dk Subject: Newbie site! From: "Thomas Mørch" <thm@eci.dk> Date: Wed, 14 Apr 1999 10:16:45 +0200 Delivered-To: mailing list sslug-novice@sslug.dk Mailing-List: contact sslug-novice-help@sslug.dk; run by ezmlm. Newsgroups: sslug.novice. Organization: SSLUG. Reply-to: sslug-novice@sslug.dk. Hejsa Så lige denne site, der blev annonceret på njlug's liste.. Tænkte at den måske kunne gøre nytte her :) http://sunsite.auc.dk/linux-newbie/ - Thomas.