



Learning Debian GNU/Linux

By Bill McCarty

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

Copyright (C) 1999 O'Reilly & Associates, Inc. Some screenshots are copyrighted by other owners and are used with permission. All other material may be distributed only subject to the terms and conditions set forth in the Open Publication License, Draft v0.4, 8 June 1999 or later (a copy of this version can be found in [Appendix F](#), and the latest version is presently available at <http://www.opencontent.org/openpub/>).



Learning Debian GNU/Linux

By Bill McCarty

1st Edition September 1999

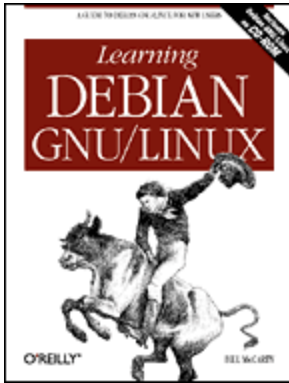
1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

About the Author

Bill McCarty is associate professor of management information systems in the School of Business and Management of Azusa Pacific University, Azusa, California, and was previously associate professor of computer science, in which capacity he taught for ten years in Azusa Pacific's Master of Applied Computer Science program.

Bill holds a Ph.D. in the management of information systems from the Claremont Graduate University, Claremont, California, and worked for 15 years as a software developer and manager.



Learning Debian GNU/Linux

By [Bill McCarty](#)

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

Colophon

David Futato was the production editor and copyeditor for *Learning Debian GNU/Linux*; Kimo Carter provided production assistance; and Claire Cloutier LeBlanc provided quality control. David Futato and Ellie Cutler proofread the text; Robert Romano and Rhon Porter created the illustrations using Adobe Photoshop 5 and Macromedia FreeHand 8. Brenda Miller wrote the index.

Hanna Dyer designed the cover layout based on a series design by Edie Freedman. Kathleen Wilson produced the cover with QuarkXPress 3.32 and Adobe Photoshop 5 software, using the ITC Garamond Condensed font. Alicia Cech designed the interior layouts, based on a series design created by Edie Freedman and Jennifer Niederst and modified by Nancy Priest. Chapter opening graphics are from the Dover Pictorial Archive and *Marvels of the New West*.

Whenever possible, our books use RepKover, a durable and flexible lay-flat binding. If the page count exceeds RepKover's limit, perfect binding is used.

Interior fonts are Adobe ITC Garamond and Adobe ConstantWillison. Text was prepared in SGML using the DocBook 2.1 DTD. The print version of this book was created by translating the SGML source into a set of *gtroff* macros using a filter developed at O'Reilly by Norman Walsh. Steve Talbott designed and wrote the underlying macro set on

the basis of the GNU *groff*-gs macros; Lenny Muellner adapted the macros to SGML and implemented the book design. The GNU *groff* text formatter version 1.09 was used to generate PostScript output; this output was distilled to PDF for use at press.

Our look is the result of reader comments, our own experimentation, and feedback from distribution channels. Distinctive covers complement our distinctive approach to technical topics, breathing personality and life into potentially dry subjects.

The cover image of a man on a bucking bull is adapted from a 19th-century engraving from *Marvels of the New West: A Vivid Portrayal of the Stupendous Marvels in the Vast Wonderland West of the Missouri River*, by William Thayer (The Henry Bill Publishing Co., Norwich, CT, 1888).



Learning Debian GNU/Linux

By Bill McCarty

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

Quotes

"O'Reilly publishes some of the best technical books anywhere. Lately they have been publishing quite a few books on Linux. The book is excellent; well worth the price - even users of non-Debian based distributions might want to pick it up!" -- William Henning, Aboutlinux.com, June 2000

"a solid book for the new Linux user considering Debian as the distribution of Linux." --Paul Gray, Cedar Valley Linux Users Group, April 2000

"If you know nothing about Linux, get this book. If the very sound of the word excites and terrifies you, all at the same time, get this book. Written primarily from an x86 viewpoint, this book is applicable to most distributions of Linux in general, and specifically to Debian on all of its supported systems. Mac users looking for a flavor of Linux to run on an old 68k, this is it. Debian runs on most 68k systems, and for that matter, it runs on most other hardware platforms. There is very little griping that can be done about this book. It's a very easy read and will get most newbies on their way to Geekdom. When it comes to getting the total newbie started with Linux, this book scores a solid 4 outta 5. It accomplishes its task quite well. Match this book with a newsgroup or a Linux mailing list, and presto - Geek in a week!" --bluebuttunny.com

"Designed to let ordinary mortals get up and running with the legendarily user-unfriendly Debian...now thanks to O'Reilly & Associates, it's within reach of Everyman." --Thom Stark, Boardwatch Magazine, March 2000

"Learning GNU/Debian Linux is about all of the awkward parts of the Debian system. The other books that have been written about the Debian distribution are not quite as helpful or comprehensive as this publication is. There aren't many books about Debian which are much better :-)" -- Richard Ibbotson, Manchester Linux Users Group, February 2000



Learning Debian GNU/Linux

By Bill McCarty

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

Copyright

Table of Contents

Preface

Chapter 1:

Why Run Linux?

Chapter 2:

Preparing to Install Linux

Chapter 3:

Installing Linux

Chapter 4:

Issuing Linux Commands

Chapter 5:

Installing and Configuring the X Window System

Chapter 6:

Using the X Window System

Chapter 7:

Configuring and Administering Linux

Chapter 8:

Using Linux Applications and Clients

Chapter 9:

Playing Linux Games

Chapter 10:

Setting Up a Linux-Based LAN

Chapter 11:

Getting Connected to the Internet

Chapter 12:

Setting Up a Linux-Based WAN

Chapter 13:

Conquering the BASH Shell

Appendix A: Linux Directory Tree

Appendix B: Principal Linux Files

Appendix C: The Debian Package Management Utilities

Appendix D: Managing the Boot Process

Appendix E: Linux Command Quick Reference

Appendix F: Open Publication License

Glossary

Index

Symbols | *A* | *B* | *C* | *D* | *E* | *F* | *G* | *H* | *I* | *J* | *K* | *L* | *M* | *N* | *O* | *P* | *Q* | *R* | *S* | *T* | *U* | *V* | *W* | *X* | *Z*



Learning Debian GNU/Linux

By Bill McCarty

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

Preface

Contents:

[Organization of This Book](#)

[Conventions Used in This Book](#)

[We'd Like to Hear from You](#)

[Acknowledgments](#)

Lately it seems that two topics crop up in conversation after conversation: the stock market and Linux. As for the stock market, I'm something of a pessimist. When friends and even perfect strangers continually recount their recent financial successes, I conclude that a stock market correction is overdue. (I've shifted my investments to bonds.)

As for Linux, I'm considerably more - perhaps wildly - optimistic. When my realtor tells me about the TV feature on Linux she saw on CNN, I see it as a harbinger of Linux Spring. Like her, my cable TV repairman, and my colleague in the next office, you've probably heard about Linux from a magazine, radio or TV program, or a friend. You're wondering what Linux is about and whether you should give it a try. If so, particularly if you currently use Microsoft Windows, this book was written for you.

Not long ago, Linux was the plaything of the technical elite. Today, however, Linux is much easier to use. Every day brings a new tool or feature designed for ease of use. If you work with Microsoft Windows and have dabbled a bit in MS-DOS, or are curious about what happens inside Windows, you can install and configure Linux. Thousands of people from all walks of life - even journalists, who are notorious for their technical ineptitude - have already done so.

This book will make your Linux journey easier, by giving you the big picture, providing you with step-by-step procedures, and getting you started doing useful or fun activities, such as word processing or

games. This book focuses on the needs of the new Linux user and on desktop Linux applications. You'll learn about networks and servers, but the details of those topics are left for more advanced books.

This book includes a CD-ROM that contains Debian GNU/Linux, so you have in your hand all you need to get started using Linux. Much of the material in this book applies to Linux generally and not merely to Debian GNU/Linux; so, even if you prefer to use another Linux distribution, you'll probably find this book useful.

Organization of This Book

Chapter 1,

Why Run Linux?, is designed to introduce you to Linux and help you determine whether Linux is appropriate for you.

Chapter 2,

Preparing to Install Linux, helps you understand what's involved in installing Linux and guides you through a procedure to gather information needed to successfully install Linux.

Chapter 3,

Installing Linux, takes you step-by-step through the installation of Linux.

Chapter 4,

Issuing Linux Commands, describes the basics of how to use the Linux command-line interface, which resembles MS-DOS but is much more powerful and sophisticated.

Chapter 5,

Installing and Configuring the X Window System, shows you how to install and configure X.

Chapter 6,

Using the X Window System, shows you how to use X, the graphical user interface included with Debian GNU/Linux. If you've used Microsoft Windows, you'll find X familiar and easy to use.

Chapter 7,

Configuring and Administering Linux, shows you how to configure your Linux system. Administering a multi-user operating system such as Linux is somewhat more complicated than administering a single-user operating system, but Linux includes tools that simplify the work.

Chapter 8,

Using Linux Applications and Clients, describes several of the most popular applications available for Linux, including desktop suites and word processors.

Chapter 9,

Playing Linux Games, describes several of the most popular games available for Linux. The chapter also shows you how to run your favorite Microsoft Windows games under Linux.

Chapter 10,

Setting Up a Linux-Based LAN, shows you how to connect your Linux system to other systems on your local area network.

Chapter 11,

Getting Connected to the Internet, shows you how to connect via your Internet Service Provider (ISP) to the Internet. Once connected, you can use your Linux system to surf the Web and access other familiar Internet services.

Chapter 12,

Setting Up a Linux-Based WAN, shows you how to set up servers that users around the world can access via the Internet. For example, you'll learn how to install and configure Apache, the world's most popular web server.

Chapter 13,

Conquering the BASH Shell, digs deeper into the BASH shell, the Linux command-line interface first introduced in **Chapter 4**. Here you'll see firsthand just how powerful and easy to use Linux can be.

Appendix A, *Linux Directory Tree*, describes the structure of the principal Debian GNU/Linux directories.

Appendix B, *Principal Linux Files*, describes the principal Debian GNU/Linux configuration files.

Appendix C, *The Debian Package Management Utilities*, describes the utilities provided by Debian GNU/Linux for working with packages. These utilities let you install applications, uninstall applications, and query a database that describes installed applications. This appendix also includes commands for installing the applications described in this book.

Appendix D, *Managing the Boot Process*, explains how PCs boot and describes how to configure your system to conveniently boot Linux.

Appendix E, *Linux Command Quick Reference*, briefly describes the most useful Linux commands. It also presents Linux equivalents for common MS-DOS commands.

Appendix F, *Open Publication License*, describes the Open Publication License this book is distributed under.

The glossary defines terms used in the book. Use it to spare yourself the effort of searching the index to discover the page on which a term is defined.

Conventions Used in This Book

The following typographical conventions are used in this book:

Boldface

indicates a keyboard command, such as **Enter**.

`Constant width`

indicates command-line computer output, code examples, and keyboard accelerators (See "Keyboard Accelerators" later in this section).

Constant width italic

indicates variables in examples.

Constant width bold

indicates user input in examples.

Italic

introduces new terms and indicates URLs or user-defined files and directories, commands, command options, file extensions, filenames, directory or folder names, and pathnames.

Path Notation

I use a shorthand notation to indicate paths. Instead of writing "Click on the Start menu, then click on Find, then Files or Folders," I write: Start → Find → Files or Folders. I distinguish menus, dialog boxes, buttons, or other GUI elements only when the context would otherwise be unclear. Simply look for the GUI element whose label matches an element of the path.

Keyboard Accelerators

In a keyboard accelerator (such as **Ctrl-Alt-Del**), a dash indicates that the keys should be held down simultaneously, whereas a space means that the keys should be pressed sequentially. For example, **Ctrl-Esc** indicates that the **Ctrl** and **Esc** keys should be held down simultaneously; whereas **Ctrl Esc** means that the **Ctrl** and **Esc** keys should be pressed sequentially.

Where a keyboard accelerator contains an uppercase letter, you should not type the **Shift** key unless it's given explicitly. For example, **Ctrl-C** indicates that you should press the **Ctrl** and **C** keys; **Ctrl-Shift-C** indicates that you should press the **Ctrl**, **Shift**, and **C** keys.

We'd Like to Hear from You

I tested and verified the information in this book to the best of my ability, but you may find that features have changed or that I've made a mistake. Please let O'Reilly know about any errors you find, by writing:

O'Reilly & Associates, Inc.
101 Morris Street
Sebastopol, CA 95472
800-998-9938 (U.S. and Canada)
707-829-0515 (International and local)
707-829-0104 (fax)

To ask technical questions or to comment on this book, please send email to bookquestions@oreilly.com.

Acknowledgments

I owe my editor, Mark Stone, a huge debt (non-monetary, I hope) for his help in carrying this book through to completion. From the initial outline to the last editorial query, Mark's suggestions and assistance were indispensable. Thanks, Mark.

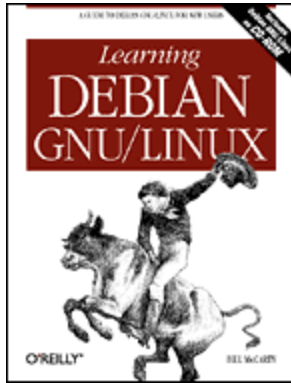
Katie Gardner helped me understand O'Reilly's authorship process, steered me clear of obstacles, and fixed many dumb mistakes without chiding me about them. Thanks, Katie.

Thanks also to Margot Maley of Waterside Productions, Inc., who brought this authorship opportunity to my attention.

Several reviewers, some working for O'Reilly & Associates and some working elsewhere, commented on the manuscript and suggested helpful corrections and improvements. In particular, members of the Debian team, especially Joey Hess, provided thorough comments on a tight schedule. I greatly appreciate their assistance and readily confess that any errors in the manuscript were added by me after their reviews, and so are entirely my responsibility.

My family - Jennifer, Patrick, and Sara - provided compassion and assistance during this latest authorship experience. Their efforts are worthy of special note, because we sold two houses and purchased a new one during the preparation of this book. They generously undertook more than their share of work on our turn of the (21st) century home, so that I could focus on writing. Thanks, guys.

I also acknowledge the love, concern, and support of my savior, Jesus Christ. His perfect love is entirely undeserved.



Learning Debian GNU/Linux

By Bill McCarty

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

1. Why Run Linux?

Contents:

[Linux at Home and at Work](#)

[What is Linux?](#)

[Reasons to Choose or Not Choose Linux](#)

[Linux Resources on the Internet](#)

This chapter introduces you to the upstart operating system Linux. It helps you determine whether Linux is right for you, by surveying the features and performance that Linux offers. It demonstrates that choosing Linux is a practical - even wise - decision for many computer users. The chapter also helps you feel at home with Linux and other Linux users, by introducing you to the history and culture of Linux. Finally, it points you to some popular gathering places on the Internet where you can correspond with other Linux users, get up-to-the-minute Linux news and information, and obtain free technical support.

1.1 Linux at Home and at Work

Perhaps you learned about Linux from a trusted friend, whose enthusiasm and ready answers convinced you to learn more about Linux, or perhaps an article or anecdote that mentioned Linux simply sparked your curiosity. In any case, you may find it interesting to learn what other computer users, ranging from PC hobbyist to guru, have accomplished by using Linux:

- Tired of slow telephone modem transfer rates, a PC owner leases a cable modem that provides high-speed transfers. He installs the new modem in a Linux system that routes packets to and from the computers of other family members. Now the entire family can simultaneously surf the Web at warp speed.
- Struggling to complete a dissertation, a graduate student determines that most of his problems stem from bugs and inadequate features of his word processing program. Dumping Microsoft Windows and Microsoft Word, he loads Linux onto his computer and uses free text processing software he downloads from the Web. In contrast to the frequent system hangs and lost work he experienced with Windows, his new system runs for over 100 days before needing to be shutdown for installation of new hardware.
- Considered among the world's best, the experienced graphics artists at Digital Domain have generated visual effects for such films as *Apollo 13*, *Dante's Peak*, *The Fifth Element*, *Interview with the Vampire*, and *True Lies*. But when director James Cameron selected Digital Domain to conjure visual effects for *Titanic*, the artists faced a task of unprecedented size and complexity. Concerned to obtain enormous computing power at the lowest cost, they purchased 160 DEC Alpha computers. Most DEC Alpha users run Microsoft Windows NT or Digital Unix as an operating system. However, Digital Domain chose to run Linux on 105 of their new computers. If you've seen *Titanic* and

Digital Domain's breathtaking effects, you know what a good decision this was.

- Needing a supercomputer, but having a budget sufficient for only a minicomputer, scientists at the Los Alamos National Laboratory created Avalon, a system of 70 networked computers that run Linux. Instead of millions of dollars, the scientists spent only about \$152,000 - none of it on software, because Linux is free. Their Avalon system performs more than 10 billion floating-point operations per second, roughly on par with the Silicon Graphics Origin2000 system, which costs \$1.8 million. Linux-based Avalon ranks as the 315th fastest computer in the world.

Linux began as a hacker's playground, but has become progressively easier to use and consequently more popular: today, perhaps as many as 7.5 million computers run Linux. Many Linux users are not hackers, but relatively ordinary computer users. Linux has become an operating system of formidable appeal and potential:

- In 1996, computing trade magazine *Infoworld* named Linux "Best Computer Desktop Operating System." A year later, they named the Linux community "Best Tech Support Organization."
- The cover of the August 10, 1998, issue of the influential business magazine *Forbes* featured super-programmer Linus Torvalds, author of the Linux kernel. The article pointed out that Intel, IBM, Netscape, Oracle, and other computing industry giants have taken a keen commercial interest in Linux and other open-source software.
- Market research firm International Data Corporation reported that in 1998, Linux held 17.2% of the server operating system market, up 212% from 1997. In contrast, Microsoft's flagship operating system, Windows NT, held a 36% market share - barely twice as great.

- Lawyers defending Microsoft against the U.S. government's antitrust charges argued that Linux poses a real threat to Microsoft's domination of the desktop operating systems market.

1.2 What is Linux?

Linux is an operating system, a software program that controls your computer. Most vendors load an operating system onto the hard drive of a PC before delivering the PC, so, unless the hard drive of your PC has failed, you may not understand the function of an operating system.

An operating system solves several problems arising from hardware variation. As you're aware, no two PC models (or models of other computers, for that matter) have identical hardware. For example, some PCs have an IDE hard drive, whereas others have a SCSI hard drive. Some PCs have one hard drive, others have two or more. Most PCs have a CD-ROM drive, but some do not. Some PCs have an Intel Pentium CPU, whereas others have an AMD K-6, and so on. Suppose that, in a world without operating systems, you're programming a new PC application, perhaps a new multimedia word processor. Your application must cope with all the possible variations of PC hardware. As a result, it becomes bulky and complex. Users don't like it because it consumes too much hard drive space, takes a long time to load, and - because of its size and complexity - has more bugs than it should.

Operating systems solve this problem by providing a single standard way for applications to access hardware devices. When an operating system exists, applications can be more compact, because they share the commonly used code for accessing the hardware. Applications can also be more reliable because this code is written only once, and by expert programmers, rather than by every application programmers.

As you'll soon learn, operating systems do many other things as well; for example, they generally provide a filesystem so that you can store and retrieve data, and a user interface so that you can control the operation of your computer. However, if you think of a computer's operating system as its subconscious mind, you won't be far off the mark. It's the computer's conscious mind - applications such as word

processors and spreadsheets - that do useful work. But, without the subconscious - the operating system - the computer would cease breathing and applications would not function.

Pronouncing Linux

Internet newsgroup participants have long debated the proper pronunciation of *Linux*. Because the name Linux was conferred by Linux kernel author Linus Torvalds, his pronunciation of the word should reign as standard as I see it. However, Linus is Finnish and his pronunciation of Linux is difficult for English speakers to approximate. Consequently, many variations in pronunciation have arisen. The most popular pronunciation sounds as though the word were spelled *Linnucks*, with the stress on the first syllable.

If your computer has a sound card, you can hear how Linus Torvalds pronounces Linux: <http://www.ssc.com/lj/linuxsay.html>.

1.2.1 PC Operating Systems

Now that you know what an operating system is, you may be wondering what operating system your PC uses. Chances are, your PC operating system was provided by Microsoft. [Table 1.1](#) shows the sales of several popular desktop operating systems during 1997 and projected sales for 2001.[\[1\]](#) Bear in mind that, because Linux is a free operating system, Linux sales are a mere fraction of Linux installations. Moreover, unlike most commercial operating systems, Linux is not sold under terms of a per-seat license; a company is free to purchase a single Linux CD-ROM diskette and install Linux on as many computer systems as they like.

[1] Source: International Data Corporation.

Table 1.1: Sales of Popular Desktop Operating Systems

--

Operating System	1997[2]	2001 (est.)[3]
Windows 95/98	69.4%	65.0%
Windows NT Workstation	9.2	26.2
DOS with Windows 3.x	7.7	0.3
MacOS	4.6	1.9
Linux	2.4	4.2
DOS without Windows	2.3	0.3
Unix	1.0	0.5
OS/2 Warp	0.8	1.2
Other	2.7	0.5

[2] U.S. sales of desktop operating systems as percent of market.

[3] Includes IBM, Digital Research (DR), and Microsoft versions of DOS.

As the table shows, your desktop computer is probably running Microsoft Windows 95 or Windows 98, which together accounted for over 69% of 1997 sales. The sales of Linux were miniscule in comparison: a mere 2.4%. As explained, these figures don't do full justice to the ubiquity of Linux. Nevertheless, notice that sales of Linux are expected to almost double, whereas those of Windows 95/98 are expected to slightly contract.

Later in this chapter you'll learn how Linux is distributed, but recall that Linux was termed a *free* operating system. If you have a high-speed Internet connection, you can download, install, and use Linux without paying anyone for anything (except perhaps your Internet Service Provider, who may impose a connection fee). It's anyone's guess how many people have downloaded Linux, but estimates indicate that between 7 and 10 million computers run Linux.

Moreover, many Linux users run Linux not as a desktop computer but as a server, which is powered up and on-online 24 hours per day, connected (at least occasionally) to the Internet, and ready to provide services to requesting clients. For example, many Linux users run web servers, hosting web sites browsed by users worldwide. But, the number of desktop Linux users - those who power on their computer to use it and power it off when they're done - is rising.

Desktop use of Linux is the focus of this book. However, if you're unfamiliar with Linux and Unix, this book is right for you even if you plan to establish a Linux server. This book will take you through the basics of setting up and using Linux. After you've mastered what this book offers, you should consult *Running Linux, Third Edition*, Matt Welsh, Matthias Kalle Dalheimer and Lar Kaufman (O'Reilly, 1999), a more advanced book that focuses on setting up and using Linux servers.

1.2.2 How Linux is Different

Linux is distinguished from many popular operating systems in three important ways.

- Linux is a cross-platform operating system that runs on many computer models. Only Unix, an ancestor of Linux, rivals Linux in this respect. In comparison, Windows 95 and Windows 98 run only on CPUs having the Intel architecture. Windows NT runs only on CPUs having the Intel architecture or the DEC Alpha.
- Linux is free, in two senses. First, you may pay nothing to obtain and use Linux. On the other hand, you may choose to purchase Linux from a vendor who bundles Linux with special documentation or applications, or who provides technical support. However, even in this case, the cost of Linux is likely to be a fraction of what you'd pay for another operating system. So, Linux is free or nearly free in an economic sense.

Second, and more important, Linux and many Linux applications are distributed in source form. This makes it possible for you and others to modify or improve them. You're not free to do this with most operating systems, which are distributed in binary form. For example, you can't make changes to Microsoft Windows or Microsoft Word - only Microsoft can do that. Because of this freedom, Linux is being constantly improved and updated, far outpacing the rate of progress of any other operating system. For example, Linux will likely be the first operating system to support Intel's forthcoming Merced 64-bit CPU.

- Linux has attractive features and performance. Free access to Linux source code lets programmers around the world implement new features, and tweak Linux to improve its performance and reliability. The best of these features and tweaks are incorporated in the standard Linux kernel or made available as kernel patches or applications. Not even Microsoft can mobilize and support a software development team as large and dedicated as the volunteer Linux software development team, which numbers in the hundreds of thousands, including programmers, code reviewers, and testers.

1.2.2.1 The origins of Linux

Linux traces its ancestry back to a mainframe operating system known as Multics (Multiplexed Information and Computing Service). Begun in 1965, Multics was one of the first multi-user computer systems and remains in use today. Bell Telephone Labs participated in the development of Multics, along with the Massachusetts Institute of Technology and General Electric.

Two Bell Labs software engineers, Ken Thompson and Dennis Richie, worked on Multics until Bell Labs withdrew from the project in 1969. One of their favorite pastimes during the project had been playing a multi-user game called Space Travel. Now, without access to a Multics computer, they found themselves unable to indulge their fantasies of flying around the galaxy. Resolved to remedy this, they decided to port the Space Travel game to run on an otherwise unused PDP-7 computer. Eventually, they implemented a rudimentary operating system they named *Unics*, as a pun on *Multics*. Somehow, the spelling of the name became *Unix*.

Their operating system was novel in several respects, most notably portability. Most previous operating systems had been written for a specific target computer. Just as a tailor-made suit fits only its owner, such an operating system could not be easily adapted to run on an unfamiliar computer. In order to create a portable operating system, Ritchie and Thompson first created a programming language, called C. Like assembly language, C let a programmer access low-level hardware facilities not available to programmers writing in a high-level language such as FORTRAN or COBOL. But, like FORTRAN and COBOL, a C program was not bound to a particular computer. Just as a ready-made suit can be lengthened or shortened here and there to fit a purchaser, writing Unix in C made it possible to easily adapt Unix to run on computers other than the PDP-7.

As word of their work spread and interest grew, Ritchie and Thompson made copies of Unix freely available to programmers around the world. These programmers revised and improved Unix, sending word of their changes back to Ritchie and Thompson, who incorporated the best such changes in their version of Unix.

Eventually, several Unix variants arose. Prominent among these was BSD (Berkeley Systems Division) Unix, written at the University of California, Berkeley, in 1978. Bill Joy, one of the principals of the BSD project, later became a founder of Sun Microsystems, which sold another Unix variant (SunOS) to power its workstations. In 1984, AT&T, the parent company of Bell Labs, began selling its own version of Unix, known as System V.

1.2.2.2 Free software

What Ritchie and Thompson had begun in a distinctly non-commercial fashion ended up spawning several legal squabbles. When AT&T grasped the commercial potential of Unix, it claimed Unix as its intellectual property and began charging a hefty license fee to those who wanted to use its Unix. Soon, others who had implemented Unix-like operating systems were distributing licenses only for a fee. Understandably, those who had contributed improvements to Unix considered it unfair for AT&T and others to appropriate the fruits of their labors. This concern for profit was unlike the democratic, share-and-share-alike spirit of the early days of Unix.

Some, including MIT scientist Richard Stallman, yearned for the return of those happier times and the mutual cooperation of programmers that then existed. So, in 1983, Stallman launched the GNU (GNU's not Unix) project, which aimed at creating a free Unix-like operating system. Like early Unix, the GNU operating system was to be distributed in source form so that programmers could read, modify, and redistribute it without restriction. Stallman's work at MIT had taught him that, by using the Internet as a means of communication, programmers the world over could improve and adapt software at incredible speed, far outpacing the fastest rate possible using traditional software development models, in which few programmers actually see one another's source code.

As a means of organizing work on the GNU project, Stallman and others created the Free Software Foundation (FSF), a non-profit

corporation that seeks to promote free software and eliminate restrictions on the copying, redistribution, understanding, and modification of software. Among other activities, the FSF accepts tax-deductible charitable contributions and distributes copies of software and documentation for a small fee, using this revenue to fund its operations and support the GNU project.

If you find it peculiar that the FSF charges a fee - even a small fee - for "free" software, you should understand that the FSF intends the word *free* to refer primarily to freedom, not price. The FSF believes in three fundamental software freedoms:

- You can copy GNU software and give it away to anyone you choose.
- If you're a programmer, you can modify GNU software any way you like, because you have access to the source code.
- You can distribute improved versions of GNU software. However, you cannot charge anyone a fee for using your improved version (although you can charge a fee for providing a user with a physical copy of your software).

1.2.2.3 Copyleft

Commercial software vendors protect their proprietary rights to software by copyrighting the software. In contrast, the FSF protects software freedom by *copylefting* its software. If the FSF placed its software in the public domain, others would be free to transform it into a proprietary product, denying users the freedom intended by the original author of the software. For example, a company might distribute the software in binary rather than source form and require payment of a license fee for the privilege of making additional copies.

To copyleft software, the FSF uses the same legal instrument used by proprietary software vendors - the copyright - but the FSF adds

special terms that guarantee freedom to users of the software. These terms, referred to as the *GNU Public License*, give everyone the right to use, modify, and redistribute the software (or any software derived from it), but only if the distribution terms are unchanged. Thus someone who attempts to transform FSF software into a proprietary product has no right to use, modify, or distribute the product. As the FSF puts it, "Proprietary software developers use copyright to take away the users' freedom; we use copyright to guarantee their freedom. That's why we reverse the name, changing *copyright* into *copyleft*."

1.2.2.4 The Linux kernel

By the early 1990s, the FSF had obtained or written all the major components of the GNU operating system except for one: the kernel. About that time, Linus Torvalds, a Finnish computer science student, began work on a kernel for a Unix-like system. Linus had been working with Minix, a Unix-like operating system written by Andrew Tannenbaum primarily for pedagogical use. Linus was disappointed by the performance of the Minix kernel and believed that he could do better. He shared his preliminary work with others on Internet newsgroups. Soon, programmers around the world were working together to extend and improve his kernel, which he called Linux (for *Linus's Minix*). As [Table 1.2](#) shows, Linux grew rapidly. Within three years of its October 5, 1991 initial release, Linux was released as production software; version 1.0 was released in March of 1994. However, as early as 1992, Linux had been integrated with other GNU software to produce a fully functional operating system, which took as its name the name of its kernel.

Table 1.2: The History of Linux

Year	Version	Users	Kernel size (Bytes)	Milestone(s)

Year	Version	Users	Kernel size (Bytes)	Milestone(s)
1991	0.01	100	63,362	Linus Torvalds writes Linux kernel
1992	0.99	1000	431,591	GNU software integrated with Linux kernel, producing a fully functional operating system
1993	0.99	20,000	937,917	High rate of code contributions prompts Linus to delegate code review responsibility
1994	1.0	100,000	1,016,601	First production release
1995	1.2	500,000	1,850,182	Linux adapted to non-Intel processors
1996	2.0	1,500,000	4,718,270	Linux supports multiple processors, IP masquerading, and Java
1999	2.2	7,500,000	10,600,000[4]	Linux growth rate exceeds that of Microsoft Windows NT

[4] estimated

However, work on Linux did not cease. Since the initial production release, the pace of development has accelerated as Linux has been adapted to include support for non-Intel processors and even multiple processors, sophisticated TCP/IP networking facilities such as IP masquerading, and more. Versions of Linux are now available for such computer models as the Apple PowerPC, the DEC Alpha, the Motorola 68k, the Sun SPARC, the Mips, and many others. Moreover, Linux does not implement an obscure Unix variant: it generally complies with the POSIX (Portable Operating System Interface) standard that forms the basis of the X/Open specifications of The Open Group.

1.2.2.5 The X Window System

Another important component of Linux is its graphical user interface, the X Window System. Unix was originally a mouseless, text-based system that used noisy teletype machines rather than modern CRT monitors. The Unix command interface is very sophisticated and, even today, some power users prefer it to a point-and-click graphical environment, using their CRT monitor as though it were a noiseless teletype. Consequently, some remain unaware that Unix long ago outgrew its text-based childhood, and now provides users a choice of graphical or command interfaces.

The X Window System (or simply X) was developed as part of the Massachusetts Institute of Technology's (MIT) Project Athena, which it began in 1984. By 1988, MIT released X to the public. MIT has since turned development of X over to the X Consortium, which released version 6 in September 1995.

X is a unique graphical user interface in two major respects. First, X integrates with a computer network, letting users access local and remote applications. For example, X lets you open a window that represents an application running on a remote server: the remote server does the heavy-duty computing; all your computer need do is pass the server your input and display the server's output.

Second, X lets you configure its look and feel to an amazing degree. To do so, you run a special application - called a *window manager* - on top of X. A variety of window managers is available, including some that closely mimic the look and feel of Microsoft Windows.

1.2.2.6 Linux distributions

Because Linux can be freely redistributed, you can obtain it in a variety of ways. Various individuals and organizations package Linux, often combining it with free or proprietary applications. Such a package that includes all the software you need to install and run Linux is called a *Linux distribution*. [Table 1.3](#) shows some of the most popular Linux distributions.

Table 1.3: Popular Linux Distributions and Their Web Home Pages

Distribution	Home Page
Caldera OpenLinux	http://www.caldera.com/
Debian Linux	http://www.debian.org/
Slackware Linux	http://www.cdrom.com/titles/os/slackwar.htm/
Red Hat Linux	http://www.redhat.com/
SuSE. Linux	http://www.suse.com/

Caldera, Red Hat, Slackware, and SuSE are packaged by commercial companies, which seek to profit by selling Linux-related products and

services. However, because Linux is distributed under the GNU GPL, you can download these distributions from the respective companies' web sites or make additional copies of a Linux distribution you purchase from them. (Note, however, that you cannot necessarily make additional copies of proprietary software that these companies may distribute with their Linux distribution.) Debian GNU/Linux is the product of volunteer effort conducted under the auspices of Software In The Public Interest, Inc., a non-profit corporation. This book is bundled with a copy of Debian GNU/Linux, which you can install and run on your PC.

1.2.3 Linux Features and Performance

The origins of Linux and the availability of its source code set it apart from other operating systems. But most users choose an operating system based on features and performance - and Linux delivers these in spades. [Table 1.4](#) compares certain features and performance characteristics of Linux with those of Microsoft Windows NT 4.0 and Sun Microsystems Solaris 2.6.[\[5\]](#) Each of these three operating systems can be run on an Intel-architecture PC.

[5] Source: Adapted from *SunWorld*, August 1998.

Table 1.4: Linux Features and Performance Comparison

Characteristic	Linux	Windows NT	Solaris
Range of compatible hardware	Very wide	Modest	Narrow
Minimal hardware	386 PC	486 PC	Pentium

Characteristic	Linux	Windows NT	Solaris
Representative cost of hardware	\$200	\$1300	\$1600
Average downtime	Very low	As low as 30 min./week	Very low
Performance	High	Comparable to Linux	Half of Linux to same as Linux
Multi-processing capabilities	Excellent	Modest	Excellent
IP Security (IPSec)	Yes	Planned	1999
IPv6	Available	Privately demonstrated	Beta
Overall user satisfaction, per <i>Datapro</i>	Highest	Lowest	Medium
Source code readily available	Yes	No	No

Characteristic	Linux	Windows NT	Solaris
Installed base	Millions	Millions	Hundreds of thousands

As you can see, Linux fares well in this comparison. It runs on a wider range of hardware platforms and runs adequately on less costly and powerful systems. Moreover, the typical downtime of a Linux system is less than that of a Windows NT system and its performance surpasses that of a Solaris system. Its multi-processing capabilities exceed those of Windows NT and its support of advanced TCP/IP networking facilities is superior to that of Windows NT and Solaris. As a group, Linux users are more satisfied than Windows NT users and Solaris users. Linux source code is readily available. And, the Linux installed base dwarfs that of Solaris and approaches that of Windows NT.

But this impressive inventory of selling points doesn't end the matter. Let's consider some other technical characteristics of Linux that distinguish it from the pack. Foremost in the minds of many is the low cost of Linux. Comparable server operating systems can cost more than \$100,000. The low cost of Linux makes it practical for use even as a desktop operating system. In that mode, it truly eclipses the competition.

Many desktop systems are occasionally, even regularly, employed as servers. Because Linux was designed for use as a server operating system, its features and performance readily outshine those of desktop operating systems used as makeshift servers. For example, Microsoft's software license for Windows NT Workstation restricts the number of simultaneous client connections to 10; if your Windows NT Workstation computer accepts more than 10 client connections, it is operating in breach of license. However, Linux

imposes no such restriction: your Linux desktop is free to accept as many client connections as you think it can handle.

Again, because it was designed as a server, Linux provides more reliable data storage than competing desktop operating systems. Most Linux users store their disk data using the EXT2 filesystem, which is superior in performance and reliability to filesystems (partition types) provided by Microsoft operating systems, including FAT, FAT32, and NTFS. Of course, Microsoft claims that its NTFS filesystem is so reliable that you'll probably never need to use special software tools to recover lost data - truth is, Microsoft provides no such tools. Despite Microsoft's ambitious claims, users report that NTFS reliability is not perfect. Here's a case in point:

When my Windows NT Workstation computer crashed a little over a year ago, I discovered that its NTFS file system was damaged. I searched the Microsoft web site for recovery instructions and tools and found nothing that helped. So, I went to my local software store and purchased a third party disk recovery tool for Windows NT. When I opened the box, I was angered to discover that it supported recovery of FAT and FAT32 data, but not NTFS data.

Eventually, I recovered 95 percent of my data by using a free Linux utility that was able to open the damaged NTFS partition and copy its files. If I'd been without Linux, I'd be without my data.

Like other server operating systems, Linux provides advanced disk management (RAID), which makes it possible to automatically duplicate stored data on several hard drives. This greatly improves the reliability of data storage; if one hard drive fails, the data can be read from another. Competing desktop operating systems such as Microsoft Windows 95/98 do not support this capability (though several third parties sell drivers that let you add this capability to your desktop operating system).

If you're an old computer dog who remembers the days of MS-DOS, you may have a fondness for what's now called the MS-DOS Prompt window. However, if you've worked exclusively within the Microsoft Windows point-and-click environment, you may not fully understand what the MS-DOS Prompt window is about. The MS-DOS Prompt window provides what's called a *command-line interface*. By typing commands, chosen from a list of commands the operating system understands, you can direct the computer to perform a variety of tasks.

For most users, the command interface is not as convenient as the point-and-click interface offered by Microsoft Windows. That's because you must know the commands the operating system understands, and must type them correctly, if you expect the operating system to do your bidding.

However, the MS-DOS Prompt window lets you accomplish tasks that would be cumbersome and time-consuming if performed by pointing and clicking. Linux comes with a similar command interface, known as the *shell*. But, the word "similar" fails to do justice to the Linux shell's capabilities, because the MS-DOS command line provides a fraction of the capabilities provided by the Linux shell.

In particular, the MS-DOS command line lacks many ease-of-use features found in the Linux shell. You may have used the MS-DOS command line and, finding it distastefully cumbersome, forever rejected it in favor of pointing and clicking. If so, you'll be pleasantly surprised to see how easy it is to use the Linux shell. You'll certainly be pleased - perhaps amazed - by the enormous power it offers. You'll learn more about the Linux shell in **Chapter 4**,

Issuing Linux Commands.

If you're a programmer, you'll also admire the ease with which it's possible to develop portable, Unix-compliant software by using Linux. Linux comes with a complete suite of software development tools, including an assembler, C compiler, C++ compiler, `make` application, and source code librarian. All of these are freely distributable programs made available under the terms of the GNU GPL.

1.3 Reasons to Choose or Not Choose Linux

Here are several reasons for running Linux. The more of these that are true of you, the likelier you are to be happy in running Linux:

You want a stable and reliable computing platform.

No other popular operating system is more stable and reliable than Linux. If you're tired of crashes and hangs and the lost time and data they entail, you're a candidate for Linux.

You want a high performance computing platform.

Linux can coax blazingly fast performance out of hardware below the minimum required to load and run other popular operating systems. And, with ample memory and a fast CPU, Linux goes toe-to-toe with anything Microsoft or other vendors offer. If speed is your thing, Linux is your hot rod.

You need a low-cost or free operating system.

If you're someone on a budget, such as a student, or if you need to set up many systems, the low cost of Linux will let you reserve your hard-earned capital for hardware or other resources. Linux is the best operating system value on the planet.

You're a heavy network or Internet user.

If you use networks, especially the Internet, Linux's advanced support for TCP/IP may light up your life. Linux makes it easy to construct firewalls that protect your system against hackers or routers that let several computers share a single network connection.

You want to learn Unix or TCP/IP networking.

The best way - perhaps the only way - to learn more about Unix or TCP/IP networking (or computers generally) is through hands-on experience. Whether you're interested in such experience owing to personal curiosity or career ambition (system administrators are often handsomely paid), Linux affords you the opportunity to gain such experience at low cost, without leaving the comfort of your home.

You seek an alternative to Microsoft's vision of computing's future.

If you're tired of marching to the relentless drumbeat of the Redmond juggernaut, Linux offers a viable way to cut the umbilical cord and set about creating a new computing destiny for yourself and others.

You want to enjoy enhanced peer esteem.

If you're a technical worker, such as a programmer or engineer, you may acquire enhanced status among your peers by being an early adopter of Linux. (Of course, in many peer groups, it's already too late to become an early adopter of Linux; but at least you won't become a late adopter). You can even obtain decals and bumper stickers to advertise your good taste in operating systems (see the Linux Mall at <http://www.all-linux.com/index.html>).

You want to have fun.

Hopefully, you've discovered that one of the best reasons for doing anything is that it's fun. Many Linux users report that they've never had so much fun with a computer. There's no better reason for running Linux than that.

To be both blunt and honest, some folks shouldn't run Linux. If one or more of the following are true of you, you should run Linux only if you have a good friend who's knowledgeable about Linux, available by phone at odd hours, and works cheap:

You're scared of computers.

If you're scared of computers, you should spend more time working with Microsoft Windows 95 or Windows 98 before venturing into the Linux world. Linux may indeed be right for you, but it's not right just yet.

You don't like to learn.

Setting up and running Linux will require you to learn new concepts and skills. None of these are especially difficult, but if you don't like to learn, setting up and running Linux will stress you out. Instead, you should stick with the familiar.

You're married to certain Windows applications.

You can run some Windows applications under Linux's WINE emulation (over 100 applications at the time of writing, Microsoft's Minesweeper and FreeCell among them). However, this isn't true of every Windows application. Before putting your toe in the Linux waters, you should obtain up-to-date information on the status of WINE emulation of your favorite Windows applications (see <http://www.winehq.com/>).

Rather than convert your desktop system to run Linux, you may prefer to install Linux on a second system or convert your existing Windows system into a dual-boot system that can run Windows or Linux. That way, you have your choice of running your favorite Windows applications or Linux.

1.4 Linux Resources on the Internet

This section points you to up-to-the-minute information about Linux available through web pages and Internet newsgroups. You may find this information helpful in completing your installation of Linux and you'll certainly find it helpful in using your Linux system.

1.4.1 Web Pages

[Table 1.5](#) lists the URLs of some popular Linux web pages. Check these out to get the latest information about Linux. Perhaps the most useful is the home page of the Linux Documentation Project. There, you can find almost anything you want to know about Linux. The Linux Documentation Project web site includes a search engine that makes it easy to find what you need.

Table 1.5: Recommend Linux Web Pages

Web page	URL
Debian Project Web Page	http://www.debian.org/
Eric S. Raymond's Linux Reading List HOWTO	http://metalab.unc.edu/LDP/HOWTO/Reading-List-HOWTO.html

Web page	URL
Gary Singleton's Gary's Place: Linux News Tips and Links	<u>http://gary.singleton.net/</u>
Joshua Go's Linux Guide	<u>http://jgo.local.net/LinuxGuide/</u>
Linux Documentation Project	<u>http://metalab.unc.edu/linux/</u>
Linux Journal Web Page	<u>http://www.linuxjournal.com/</u>
Linux Journal's Linux Gazette	<u>http://www.linuxgazette.com/</u>
Linux Resources	<u>http://www.linuxresources.com/</u>
Linux Web Ring	<u>http://nll.interl.net/lwr/</u>
Linux Weekly News	<u>http://lwn.net/</u>

Web page	URL
O'Reilly & Associates Linux Center	http://linux.oreilly.com/
Renaisssoft's Linux Resources	http://www.renaisssoft.com/linux.html
Robert Kiesling's Linux Frequently Asked Questions with Answers (FAQ)	http://metalab.unc.edu/LDP/FAQ/Linux-FAQ.html
Slashdot	http://slashdot.org/
Victoria, British Columbia Linux Users Group	http://vlug.org/vlug/

The Linux Webring offers another convenient way to explore a variety of Linux-related web sites. Participating web sites present links to one another; by following these links you can circumnavigate the entire ring or you can use the Webring's home page to seek exactly the sort of page you're interested in.

Linux Journal is a popular magazine among Linux users. You can subscribe to the hard copy edition or peruse any of several web sites supported by *Linux Journal*.

A FAQ (Frequently Asked Questions) list summarizes questions and answers commonly appearing on an Internet newsgroup or other venue. FAQs are among the most valuable sources of information about any topic because they answer a high percentage of potential questions. You should consult FAQs early and often.

The motto of the Slashdot web site is "News for nerds. Stuff that matters." You'll find a great deal of interesting news and information there, concerning not only Linux but the open source community and computing generally.

1.4.2 Newsgroups

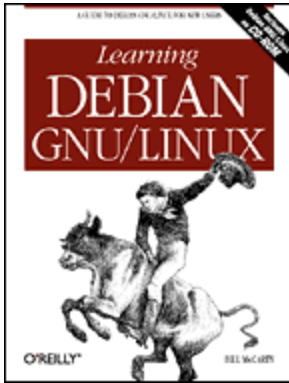
Internet newsgroups are a popular gathering place for Linux users. There, they give and receive help in setting up and using Linux systems and share pointers to the latest Linux software. [Table 1.6](#) lists some popular Linux-related newsgroups. If your Internet Service Provider supports access to newsgroups, you can view them using Microsoft Internet Explorer, Netscape Communicator, or a special newsgroup client application.

Table 1.6: Popular Linux-related Internet Newsgroups

Newsgroup	Topic
comp.os.linux.advocacy	Arguing the benefits of Linux in comparison to other operating systems
comp.os.linux.alpha	Linux on DEC Alpha computers

Newsgroup	Topic
<u>comp.os.linux.announce</u>	Announcements important to the Linux community (Also visit the archive site)
<u>comp.os.linux.answers</u>	FAQs, HOWTOs, READMEs, etc.
<u>comp.os.linux.development.apps</u>	Writing Linux applications and porting applications to Linux
<u>comp.os.linux.development.system</u>	Linux kernels, device drivers, and modules
<u>comp.os.linux.hardware</u>	Hardware compatibility with the Linux operating system
<u>comp.os.linux.misc</u>	Topics not covered by other groups
<u>comp.os.linux.networking</u>	Networking and communication.
<u>comp.os.linux.powerpc</u>	Linux systems running on the PowerPC

Newsgroup	Topic
<u>comp.os.linux.setup</u>	Linux installation and system administration
<u>comp.os.linux.x</u>	X servers, clients, libs, and fonts



Learning Debian GNU/Linux

By Bill McCarty

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

2. Preparing to Install Linux

Contents:

[Minimum Hardware Requirements](#)

[Collecting Information About Your System](#)

[Preparing Your Hard Disk](#)

This chapter presents information you need to know and tasks you need to perform before installing Linux. It helps you make certain that your IBM-compatible PC meets the minimum hardware requirements for Linux. It shows you how to document your system configuration so that you can respond to questions presented by the Linux install procedure. Finally, it shows you how to prepare your hard disk for Linux.

2.1 Minimum Hardware Requirements

Linux supports a wide range of PC hardware; but not even Linux supports every known device and system. Your PC must meet certain minimum requirements in order to run Linux. The following sections present these minimum requirements; however, for the latest and most complete information, you should check the Debian Project web site at <http://www.debian.org/>. The Debian web site will also help you determine if Linux supports all the devices installed in your system.

2.1.1 Central Processing Unit (CPU)

Linux does not support the Intel 286 and earlier processors. However, it fully supports the Intel 80386, 80486, Pentium, Pentium Pro, Pentium II, and Pentium III processors. Nevertheless, some users feel that their 80386 Linux systems respond sluggishly, particularly when running X. So, if you want optimum performance, you should install Linux on a PC having an 80486 processor or better.

Linux also supports non-Intel processors such as the Cyrix 6x86 and the AMD K5 and K6. Most Linux users have systems that use Intel chips; if your system uses a non-Intel chip, you may find it more difficult to resolve possible problems.

2.1.2 Motherboard

Linux supports the standard ISA, EISA, PCI, and VESA (VLB) system buses used on most IBM-compatible PCs. Linux recently gained support for IBM's MCA bus, used in IBM's PS/2 series of computers. However, at the time of this writing, Debian GNU/Linux does not yet support the MCA bus. If you have an IBM PS/2, you may be unable to install Debian GNU/Linux (check the Debian Project web site for the latest available information on support for the MCA bus).

Your motherboard should include at least 16 MB of RAM for optimum Linux performance. Some users have managed to coax Linux into working on systems with as little as 4 MB of RAM. However, if your system has less than 16 MB of RAM, you probably won't be pleased with its performance. If you plan to run X, you may wish to install more than 16 MB of RAM - perhaps 64 MB. Although X operates well with 16 MB of RAM, you can open more windows and switch between them more quickly if you have additional memory.

A handful of motherboards presents special problems when installing Linux. Generally, the problem stems from a bad BIOS, for which a fix is often available. Check the Debian Project web site for details.

2.1.3 Drives

An anonymous wag once quipped that one can never be too thin, too rich, or have too much hard disk space. Fortunately, Linux is not too hungry for disk space. To install and use Linux, you should have at least 250 MB of free hard disk space. (The minimum is about 100 MB, but installing Linux on a system with so little disk space will compel you to omit many useful applications and will leave you with little room to work.)

More realistically, if you plan to use your Linux system as a workstation, you should have at least 600 MB of free disk space; if you plan to use your Linux system as a server, you should have at least 1.6 GB (1,600 MB) of free disk space.

For convenient installation using the CD-ROM included with this book, your system should include an IDE or SCSI CD-ROM drive. It's also possible to install Linux from a PCMCIA CD-ROM drive, an FTP site, an NFS server, an SMB shared volume, or a hard drive. Consult the Debian Project web site for details.

Your system should also include a 3.5-inch floppy drive. You'll use the floppy drive to boot your system from a special Linux diskette

you create.

2.2 Collecting Information About Your System

In order to be able to complete the installation procedure smoothly, you should collect certain information about your system before beginning the installation. Often the installation utility will be able to determine your system configuration automatically but when it fails to do so, you must be prepared to supply the needed information. Otherwise, you'll be forced to terminate the installation procedure, obtain the information, and restart the installation.

2.2.1 Information You Need

[Table 2.1](#) specifies the configuration information you need. To obtain this information, you can consult your system documentation and the documentation for any devices installed by you. If your documentation is missing or incomplete, you may need to contact your hardware vendor or manufacturer. Alternatively, you may be able to find the needed information on the manufacturer's web site; use a search engine such as Yahoo! or AltaVista to discover the URL of the web site.

Table 2.1: Configuration Information Needed to Install Linux

Device	Information needed
Hard Drive(s)	<p>The number, size, and type of each hard drive.</p> <p>Which hard drive is first, second, and so on</p> <p>Which adapter type (IDE or SCSI) is used by each drive.</p> <p>For each IDE drive, whether or not the BIOS is set for LBA mode</p>

Device	Information needed
RAM memory	The amount of installed RAM
CD-ROM Drive(s)	<p>Which adapter type (IDE, SCSI, or other) is used by each drive</p> <p>For each drive using a non-IDE, non-SCSI adapter, the make and model of the drive</p>
SCSI Adapter (if any)	The make and model of the card
Network Adapter (if any)	The make and model of the card
Mouse	<p>The type (serial, PS/2, or bus)</p> <p>The protocol (Microsoft, Logitech, MouseMan, etc.)</p> <p>The number of buttons</p> <p>For a serial mouse, the serial port to which it's connected</p>
Video Adapter	<p>The make and model of the card</p> <p>The amount of video RAM</p>

To obtain the needed information, you may need to examine your system's BIOS settings or open your system's case and examine the installed hardware. Consult your system documentation to learn how to do so.

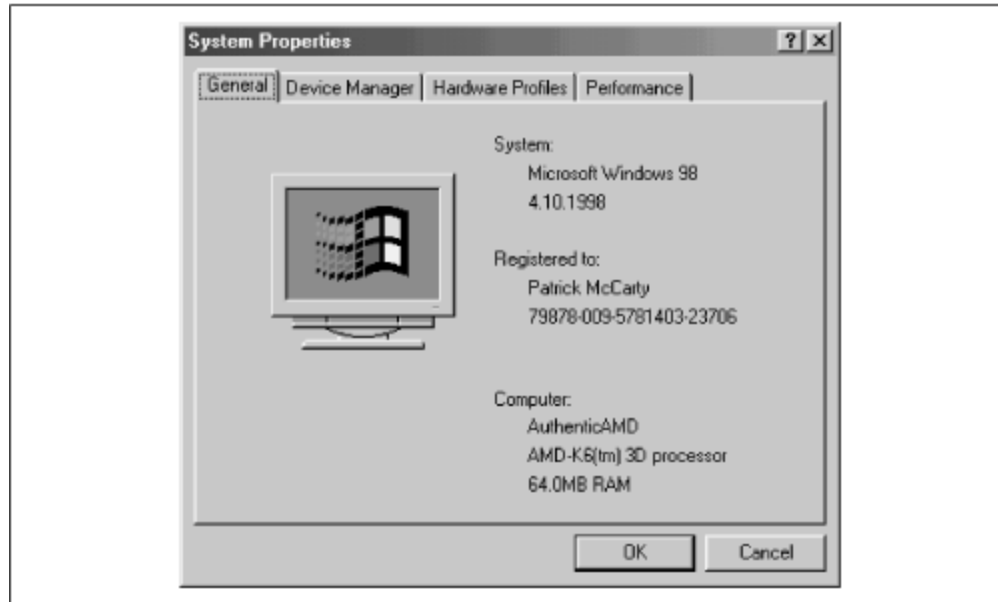
2.2.2 Collecting Configuration Information by Using Windows

If you run Microsoft Windows 95 or Windows 98, you can obtain much of the needed information by using the Windows System Properties dialog box, which you can launch by using the Control Panel:

1. Click on the Start menu. A popup menu appears.
2. Select Settings on the popup menu and click on Control Panel in the submenu. The Control Panel appears.
3. Double click on System. The System Properties dialog box appears. If necessary, click on the General tab, so that the dialog box resembles [Figure 2.1](#).

The General tab of the System Properties dialog box shows the type of your system's processor and the amount of installed RAM.

Figure 2.1: The General tab of the System Properties dialog box

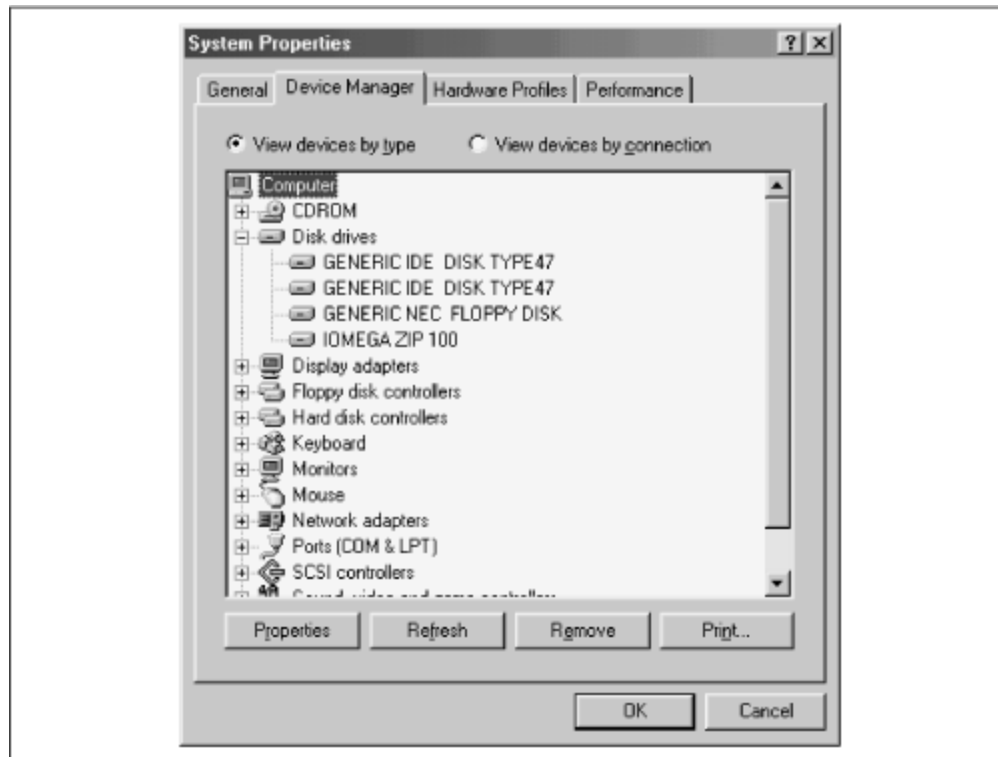


4. Click on the Device Manager tab. The appearance of the dialog box changes to resemble [Figure 2.2](#).

You can double click on an icon (or single click on the plus key adjacent to an icon) to obtain additional information. For example, by double clicking on the Disk Drives icon you can determine whether a disk drive uses an IDE or SCSI interface.

If you have a printer, you can use the Print button to print information about your system's devices.

Figure 2.2: The Device Manager tab of the System Properties dialog box



From the Device Manager tab, you can learn the following information:

- The number and type (IDE or SCSI) of your system's hard drives.
- The make and model of CD-ROM drives.

Some installed CD-ROM drives do not appear in the Device Manager tab of the System Properties dialog box. Often the *CONFIG.SYS* file will contain clues that help you learn more about such drives.

- The type of mouse installed.
- The make and model of the video adapter.
- The make and model of multimedia adapters, such as sound cards, if any.
- The make and model of network adapters, if any.

- The make and model of SCSI adapters, if any.

2.3 Preparing Your Hard Disk

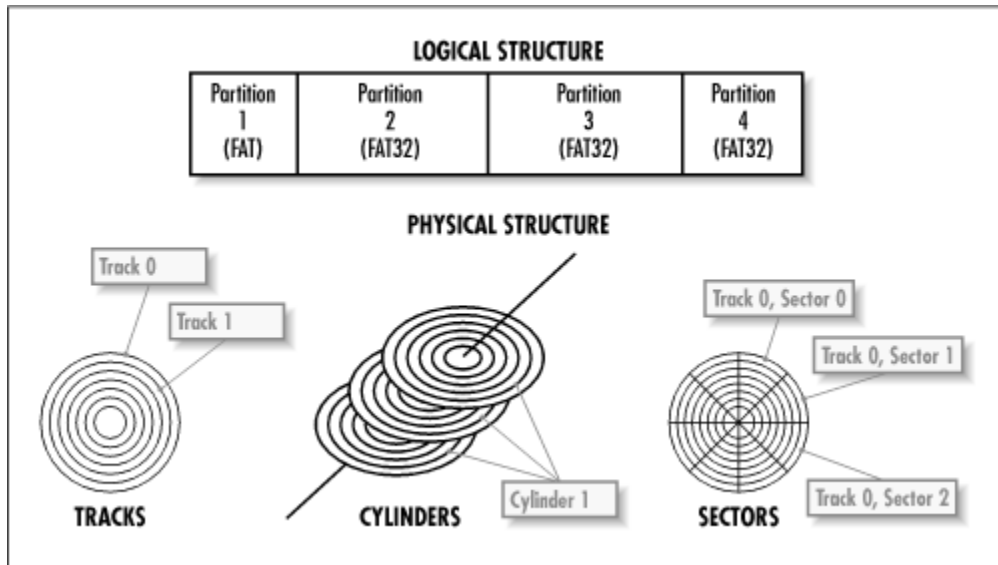
To prepare your hard disk for installing Linux, you must allocate the space in which Linux will reside. You'll learn how to do so in this section. First, you'll learn how hard disks are organized, then you'll learn how to view the structure of a hard disk. Finally, you'll learn how to alter the structure of a hard disk.

2.3.1 How Hard Disks are Organized

Let's start by reviewing facts you've probably learned by working with Microsoft Windows. Most operating systems, including Microsoft Windows 95 and Windows 98, manage hard disk drives by dividing their storage space into units known as *partitions*. So that you can access a partition, Windows 95 and Windows 98 associate a drive letter (such as C: or D:) with it. Before you can store data on a partition, you must *format* it. Formatting a partition organizes the associated space into what is called a *filesystem*, which provides space for storing the names and attributes of files as well as the data they contain. Microsoft Windows supports several types of filesystems, such as FAT and FAT32, a newer filesystem type that provides more efficient storage, launches programs faster, and supports very large hard disk drives.

Partitions comprise the *logical structure* of a disk drive, the way humans and most computer programs understand the structure. However, disk drives have an underlying *physical structure* that more closely resembles the actual structure of the hardware. [Figure 2.3](#) shows the logical and physical structure of a disk drive.

Figure 2.3: The structure of a hard disk



Mechanically, a hard disk is constructed of platters that resemble the phonograph records found in a old-fashioned juke box. Each platter is associated with a read/write *head* that works much like the read/write head on a VCR, encoding data as a series of electromagnetic pulses. As the platter spins, the heads record data in concentric rings known as *tracks*, which are numbered beginning with zero. A hard disk may have hundreds or thousands of tracks.

All the tracks with the same radius are known as a *cylinder*. Like tracks, cylinders are numbered beginning with zero. The number of platters and cylinders of a drive determine the drive's *geometry*. Most PCs require you to specify the geometry of a drive in the BIOS setup.

Most operating systems prefer to read or write only part of a track, rather than an entire track. Consequently, tracks are divided into a series of *sectors*, each of which holds a fixed number of bytes, usually 512.

To correctly access a sector, a program needs to know the geometry of the drive. Because it's sometimes inconvenient to specify the geometry of a drive, some PC BIOS programs let you specify *logical block addressing* (LBA). LBA sequentially numbers sectors, letting programs read or write a specified sector without the burden of specifying a cylinder or head number.

2.3.2 Viewing Partition Information

The first step in preparing your hard disk is viewing its partition information. Once you know how your hard disk is organized, you'll be able to determine how to reorganize it to accommodate Linux. To view the partitions that exist on your hard disk drives, you can use the `fdisk` utility:

1. Click on the Start menu. The Start popup menu appears.
2. Select Programs. The Programs submenu appears.
3. From the Programs submenu, click on MS-DOS Prompt. An MS-DOS Prompt window appears.
4. Type `fdisk` and press **Enter**. The `fdisk` menu appears, as shown in [Figure 2.4](#).

The `fdisk` menu may not appear immediately. Instead, Windows may ask if you want to enable large disk support. If this occurs, type `N` and press **Enter**. You don't need to enable large disk support to view partition information.

Figure 2.4: The `fdisk` Options screen

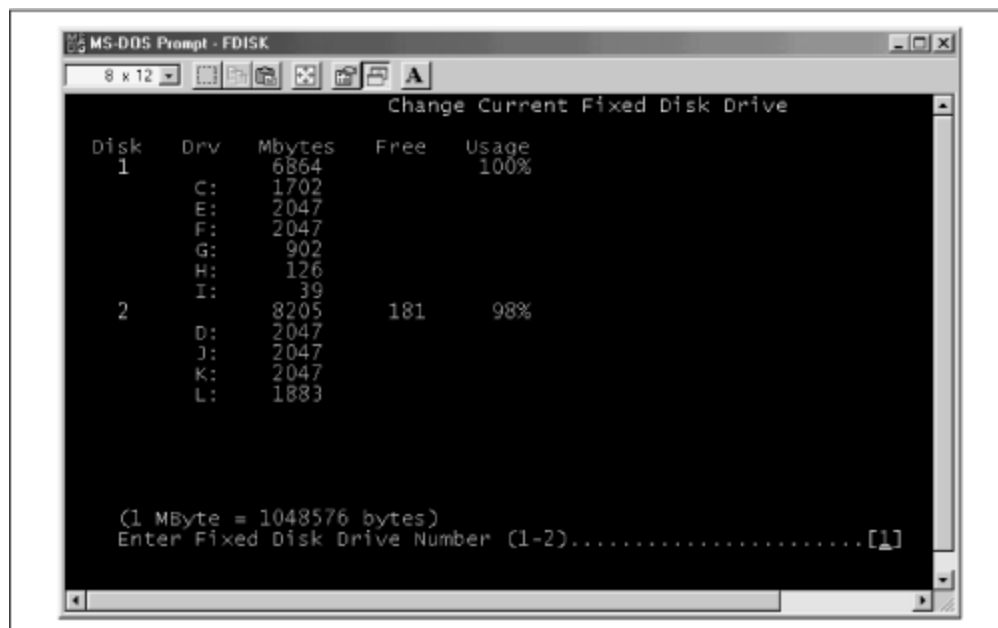


5. Type 5 and press **Enter**. This takes you to a screen, resembling the one shown in [Figure 2.5](#), that lets you specify the current fixed disk drive. This screen displays partition information in a more readable format than the screen you obtain by using menu item 4, "Display Partition Information."

The screen shows each hard disk drive and its size, numbering the drives beginning with 1. If a drive contains free space not allocated to a partition, the screen shows the amount of free space available. The screen also shows how much of the drive's space has been allocated to partitions, as a percentage of the total drive space.

Under the information describing a drive, the screen shows the size of each partition that resides on the drive. The screen also shows the associated drive letter, if any.

Figure 2.5: The fdisk Change Current Fixed Disk Drive screen



6. When you're done viewing partition information, press **Esc** twice to exit `fdisk` and return to an MS-DOS prompt. You can then

close the MS-DOS Prompt window by clicking on the close icon in the upper right corner of the window or by typing `exit` and pressing **Enter**.

2.3.3 Obtaining Sufficient Disk Space

You cannot install Linux to a partition already in use. By viewing the partitions on your hard drive, you can determine which of the following two cases best describes your system:

- You have available free (unpartitioned) disk space large enough to accommodate Linux (600 MB to 1.6 GB, depending on the type of installation you want).

In this case, make a note of the drive that holds the free disk space. You can then begin the installation process described in **Chapter 3**,

Installing Linux. However, see the following tip on PC BIOS limitations.

- You don't have enough free (unpartitioned) disk space to accommodate Linux.

The procedures given in this section will help you obtain the necessary free space.

If you don't have sufficient disk space, you have several options:

- If your system has room for an additional disk drive, you can install a new drive and use it to hold Linux. The section titled "[Section 2.3.3.1, "Installing a new disk drive"](#)" offers some considerations and tips on installing a new drive.
- If you have one or more unneeded partitions, you can delete them and use the space you gain to hold Linux. The section titled "[Section 2.3.3.2, "Identifying an unused partition"](#)" shows you how to identify an unused partition.
- If you have one or more partitions that are larger than needed, you can shrink them and use the space you gain to hold Linux. The section titled "[Section 2.3.3.3, "Shrinking a partition"](#)" shows you how to determine whether a partition is larger than needed and how to free the excess space.

2.3.3.1 Installing a new disk drive

Often, the easiest way to install Linux is to install a new disk drive. If your system has only a single hard disk drive, you can probably install a second drive and place Linux on the new drive. Before purchasing a drive, you should make sure that the system provides room to mount the new drive and that you have the proper data and power cables. You'll also need to plan how to move data from your existing hard drive to the new hard drive. Consult your system vendor for assistance, if necessary.

If your system already has two disk drives, you probably can't simply add a third disk drive: the BIOS of most PCs lets you boot the system from only the first or second hard drive. In such a case, you can probably replace one of your existing drives with a larger drive adequate to support your existing needs and Linux.

2.3.3.2 Identifying an unused partition

You can use the drive letter information provided by `fdisk` to examine the contents of a partition in the Windows Explorer. If you can find a partition that holds no useful data but that is large enough to accommodate the type of Linux installation you want, you can delete the partition and use the free space to hold Linux.

The easiest way to delete a partition is to use Debian's `cfdisk` utility. Make note of the partition you wish to delete and then simply begin the installation process described in the next chapter.

2.3.3.3 Shrinking a partition

Even if all of your partitions contain useful data, one or more partitions may be larger than required. In that case, you can reduce the size of each such partition and reorganize the drive to include contiguous unused space that you can use to hold Linux.

You can use the Windows Explorer to determine the amount of free disk space in a partition. Simply right click on the drive icon and click on Properties in the popup menu. The Properties dialog box that appears shows the amount of used and free disk space associated with the drive.

If you are able to find one or more partitions that have sufficient free space for a Linux installation, you can use a special utility to split the used and unused portions of a partition into separate partitions. The Linux CD-ROM includes the GPL `fips` utility, which can split FAT and FAT32 partitions. For information on using `fips`, see the next section.

WARNING: If you make a mistake while attempting to shrink a partition or if the software malfunctions, you may lose all data in one or more partitions. You should not attempt to shrink a partition until you've completely backed up your system and made sure that your backup is usable.

Many Linux users find PowerQuest's PartitionMagic utility helpful. Unlike `fips`, PartitionMagic is commercial software; however, it supports partition types and operations not supported by `fips`. For example, PartitionMagic can split NTFS, HPFS, and Linux `ext2` partitions.

2.3.3.4 Using the `fips` utility

The `fips` utility lets you split a FAT partition into two partitions, one containing the data of the original partition and the other containing no data. Version 2 of the `fips` utility lets you split a FAT or FAT32 partition. Once you've run `fips`, you can use the `fdisk` program to delete the new empty partition, creating free space for installing Linux.

The `fips` utility will not split a partition unless there is at least about 10 MB of free space at the end of the drive. Moreover, `fips` requires a free entry in the disk's partition table; it will not work if your drive already contains four partitions.

This section describes the procedure for using `fips`. It assumes that you're running Microsoft Windows 9x. If you're running another operating system, consult the `fips` documentation for special instructions.

WARNING: In the words of its author, `fips` is "somewhat experimental." Neither the author of this book nor the publisher can accept responsibility or liability for damage resulting from your use or misuse of `fips`. You should not attempt to use `fips` until you've completely backed up your system and made sure that your backup is usable.

Also, your Microsoft operating system may assign different letters to drives after you use `fips` to split a partition. For example, your D: drive may become E:. The `fips` utility ensures that the C: drive remains C: so that you will generally be able to boot your system; however, you may not be able to properly access programs or files that reside on drives other than C:.

Before running `fips`, you should check the condition of your hard drive by running `chkdsk`, ScanDisk, Norton Disk Doctor, or a similar program. To launch the ScanDisk program, click Start → Programs → Accessories → System Tools → ScanDisk. If your program reports errors, you should not attempt to split the partition until you resolve them.

Next, you must defragment your hard drive. Defragmenting a drive moves all its data to the beginning of the drive, leaving all the free space at the end. You can defragment your drive by using the Microsoft `defrag` utility. Simply click Start → Programs → Accessories → System Tools → Disk Defragmenter. However, you can use another defragmentation program if you prefer; the Norton Speedisk program, PCTool's Compress program, and various shareware programs are suitable.

The Microsoft `defrag` program doesn't always defragment a drive as thoroughly as possible. It sometimes erroneously regards some disk blocks as bad or immovable, and thus can fail to clear space that another program would successfully reclaim. If you find the results of using `defrag` disappointing, you should consider using a different program.

Next, you should disable virtual memory. Launch the Control Panel by clicking Start → Settings → Control Panel. Then, double click on the System icon. The System Properties dialog box appears. Select the Performance tab and click on "Virtual Memory..." The Virtual Memory dialog box appears. Make a note of the current setting. Then, click on "Let me specify my own virtual memory settings" and then click on "Disable virtual memory." Click on OK to dismiss the Virtual

Memory dialog box. Finally, click on OK to dismiss the System Properties dialog box.

Next, create a boot floppy, by using the Add/Remove Programs control panel applet. Double click on the Add/Remove Programs icon in the Control Panel. The Add/Remove Program Properties dialog box appears. Click on the Startup Disk tab and then click the Create Disk button. A progress bar appears on the Add/Remove Program Properties dialog box. When prompted by the program, insert your Windows 9x CD-ROM. After reading from the CD-ROM, the program will prompt you to insert a formatted floppy disk into your system's floppy drive. Label a floppy disk "FIPS" and insert it into the drive. As the boot disk is being written, the progress bar informs you of the task's status. After a few minutes, the progress bar will disappear, informing you that the boot disk has been created. Click on OK to dismiss the Add/Remove Program Properties dialog box.

Do not remove the diskette from the drive. Instead, copy the following files from the CD-ROM onto the floppy disk:

`\dosutils\fips20\restorrb.exe`

`\dosutils\fips20\fips.exe`

`\dosutils\fips20\errors.txt`

If you use IMAGE or MIRROR or if your *config.sys* or *autoexec.bat* file invokes programs that write to your hard disk, use the Windows Explorer to temporarily rename *config.sys* to *config.fip* and *autoexec.bat* to *autoexec.fip*. If you're unsure what programs your *config.sys* and *autoexec.bat* files invoke, play it safe by renaming both files.

Now, boot your system by using the floppy diskette you created. When the MS-DOS command prompt appears, type `fips` and press **Enter** to launch the `fips` utility. If you have more than one hard disk drive, `fips` asks which disk it should access. Respond by identifying the appropriate disk drive.

Next, `fips` gives you the opportunity to create a backup file on your A: drive. You should allow `fips` to create the file. Then, if something goes wrong in using `fips`, you can boot from your floppy diskette and run the `restorrb` program to return your hard drive to its original state.

The `fips` utility then displays the partitions found on your hard disk. You need pay attention to only the first and last columns of the display, which indicate the number and size of each partition.

The `fips` utility performs some analysis of your hard drive. Then, if your hard drive contains more than one partition, `fips` asks you which partition you wish to split. Type the number of the partition and press **Enter**.

After performing some further analysis, `fips` asks you to enter the number of the cylinder on which the new partition should begin. Use the left cursor key to decrease the number and the right cursor key to increase it. As you increase or decrease the cylinder number, `fips` displays the size of the two partitions it will create. After setting the proper cylinder number, press **Enter**.

You may find that the maximum size of the empty partition is much smaller than you expected. If so, this is probably due to the presence of a hidden file that your defragmentation program was unable to move.

To identify such files, open an MS-DOS Prompt window, type the command `dir /a:h /s` and press **Enter**. Ignore any files with names similar to *ibmbio.com* or *ibmdos.com*. Try to determine what program created any remaining hidden files. If you can identify the program, you may be able to create a larger empty partition by uninstalling the program, splitting the partition, and reinstalling the program.

The `fips` utility displays the new partition information. You can type Y to save your changes and exit, or type C to make additional changes.

After exiting `fips`, you should *immediately* boot Windows 9x and run ScanDisk to verify that the partitions created by `fips` are valid. *Do not write anything to the disk before rebooting*; otherwise, you may destroy information on your hard drive.

Next, you should re-enable virtual memory. To do so, launch the Control Panel by clicking Start → Settings → Control Panel. Then, double click on the System icon. The System Properties dialog box appears. Select the Performance tab and click on Virtual Memory. Return the settings to the values you earlier noted, then click on OK to dismiss the Virtual Memory dialog box. Then, click on OK to dismiss the System Properties dialog box.

If you renamed your `config.sys` and `autoexec.bat` files, restore the original names by using Windows Explorer.

Finally, reboot your system so that the changes to your system's virtual memory settings become active. Now you're ready to install Linux to the new empty partition.



Learning Debian GNU/Linux

By Bill McCarty

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

3. Installing Linux

Contents:

[Installing the Operating System and Applications](#)
[Getting Help](#)

In this chapter, you'll learn how to install Linux by following a simple, step-by-step procedure. Most users will be able to complete the installation procedure without difficulty; however, the chapter includes a section that describes how you can obtain help if you encounter installation problems. Once you successfully complete the installation procedure, you'll have your own working Linux system.

3.1 Installing the Operating System and Applications

To install Linux, you follow a simple, step-by-step procedure that has three main phases:

- Installing the operating system kernel and base system
- Configuring the new Linux system
- Installing applications

WARNING: Although the Linux installation procedure is generally troublefree, errors or malfunctions that occur during the installation of an operating system can result in loss of data. You should not begin the installation procedure until you have backed up all data on your system and determined that your backup is error-free.

3.1.1 The Installation Program User Interface

Like other modern Linux distributions, Debian GNU/Linux includes a screen-based install program that simplifies the installation and initial configuration of Linux. However, the install program works somewhat differently than a typical Microsoft Windows application. For instance, it does not support use of a mouse or other pointing device; all input is via the keyboard. So that you can make effective use of the install program, the next three subsections describe the user-interface controls used by the install program, present the special keystrokes recognized by the install program, and explain the use of Linux's virtual consoles.

3.1.1.1 User-interface controls

[Figure 3.1](#) shows a typical screen displayed by the install program. This screen includes the following controls:

A main window

The install program runs in a full screen window. The top line of the window displays the name of the current installation step. In [Figure 3.1](#), the current step is "Select CD Interface Type." You cannot minimize or change the size of the install program's main window.

The cursor

Like Windows programs, the installation program has a cursor on screen. Unlike Windows, the cursor movement and cursor actions are controlled by the keyboard, rather than by the mouse. The location of the cursor is called the input focus. At any time, exactly one control has the *input focus*, which lets it respond to keyboard input. The install program displays a rectangular blue cursor that identifies the field having the input focus. In [Figure 3.1](#), the list item named `/dev/hdc` has the input focus.

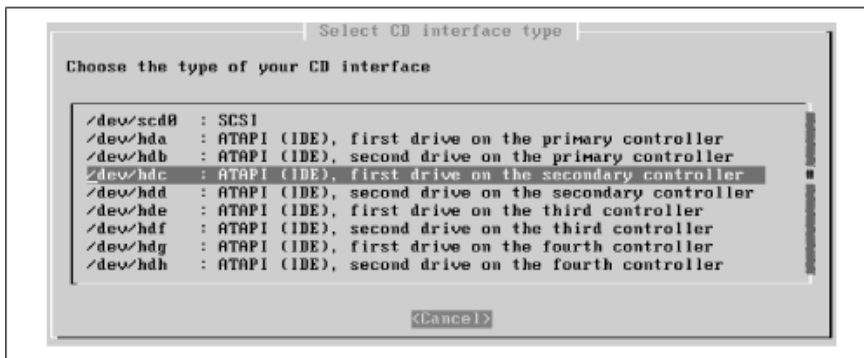
A scrollable list

Scrollable lists let you page through a list of items that may be too long to display all at once. At any time, one line in the scrollable list is active, as indicated by blue highlighting. When a scrollable list has the input focus the Up and Down arrow keys let you choose a different active item. Some scrollable lists associate actions with items; you can initiate the action associated with the active item by pressing **Enter**.

Buttons

Many install program windows include one or more buttons. You can make a button active by pressing the **Tab** key to move the cursor to the button. When a button is active, pressing **Enter** initiates the action associated with it.

Figure 3.1: A typical screen displayed by the install program



Although [Figure 3.1](#) does not show a text box, some install program windows include one. Text boxes let you type text that is sent to the install program when you press the Ok button. You can recognize a text box by the underscores that define its input area.

3.1.1.2 Common keystrokes

Several keystrokes let you direct the operation of the install program. For example, you can use the **Tab** key to move the input focus from one control to the next. [Table 3.1](#) summarizes the keystrokes that the install program recognizes. You may want to keep this table handy as you work with the install program.

Table 3.1: Keystrokes Recognized by the Install Program

Keystroke	Meaning
Enter	Send a button press to the install program.
Tab	Move the input focus to the next field.
Down	Move the cursor down.
Up	Move the cursor up.
Left	Move the cursor left.
Right	Move the cursor right.

WARNING: You should press keys only when an installation program dialog box is active. Pressing keys at other times can send keystrokes to programs invoked by the install program, which may interpret your input in an unpredictable fashion.

3.1.1.3 Using virtual consoles

A console is a combination of a keyboard and a display device, such as a video monitor. A console provides a basic user interface adequate to communicate with a computer: you can type characters on the keyboard and view text on the display device.

Although a home computer system seldom has more than one console, Linux systems provide several virtual consoles. By pressing a special combination of keys, you can control which console your system's keyboard and monitor are connected to. [Table 3.2](#) describes the virtual consoles used by the install program. The main installation dialog appears in virtual console 1. The contents of other virtual consoles can be useful in troubleshooting; however, you will not usually need to switch from one virtual console to another. Nevertheless, you may find it interesting to view the contents of the virtual consoles.

Table 3.2: Virtual Consoles Used by the Install Program

Console	Keystroke	Contents
1	ALT-F1	The installation dialog.
2	ALT-F2	A shell prompt, which lets you enter commands to be processed by Linux.
3	ALT-F3	The installation status log, containing termination messages of launched programs.
4	ALT-F4	The installation log, containing messages from the install program.

3.1.2 Installing the Kernel and Base System

If your system can boot from a CD-ROM, you can boot Linux directly from the CD that accompanies this book, which is by far the simplest way to boot Linux. If your system supports booting from a CD-ROM, configure your system to do so and boot Linux now.

3.1.2.1 Booting from MS-DOS or Windows 9x

If your system can't boot from a CD-ROM, you can boot Linux by first booting MS-DOS or Windows 9x. To do so, use File Manager to copy the following files

from the *install* directory of the CD-ROM that accompanies this book, to your Windows desktop:

boot.bat
linux
loadlin.exe
root.bin

Next, right click on the *boot.bat* file on your desktop - not the one on the CD-ROM - and select the Create Shortcut menu item. Windows creates a desktop icon named *Shortcut to boot.bat*. Right click on this icon and select the Properties menu item. A Properties dialog appears. Click on the Program tab and then click on the Advanced button. Click on the check box marked "MS-DOS mode" and then click on OK. Finally, click on OK to exit the Properties dialog.

To boot Linux, double click on the *Shortcut to boot.bat* desktop icon. A dialog box asks if you want to close all other programs and continue. Close any important applications and then click on Yes to boot Linux.

3.1.2.2 Booting from floppy diskettes

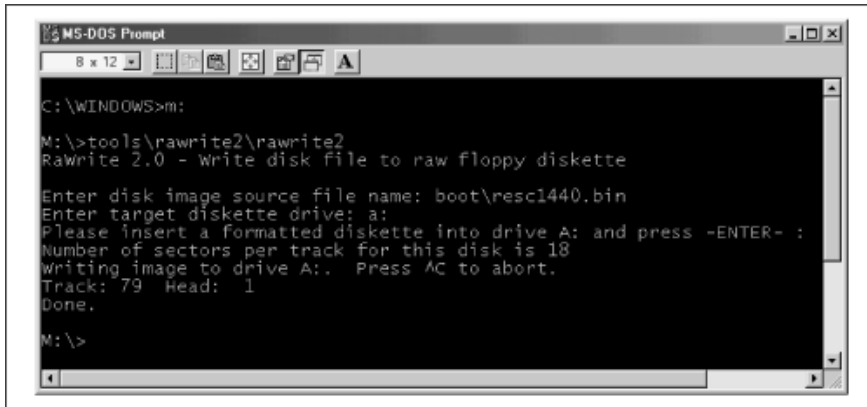
If your system can't boot from a CD-ROM diskette and you have difficulty booting Linux from MS-DOS or Windows 9x, you can boot Linux from floppy diskettes. Before beginning the installation, obtain two floppy disks. You'll use one to create the Linux installation disk and another from which to boot your Linux system.

To begin installing Linux, you must boot your system from a floppy diskette containing the boot kernel. Creating the boot disk requires some special measures; you can't simply copy files onto a disk and then boot it.

To create the boot disk, perform the following steps:

1. Insert the Linux CD-ROM in your CD-ROM drive.
2. Start an MS-DOS Prompt window by clicking on Start, selecting Programs, and clicking on MS-DOS Prompt.
3. In the MS-DOS window, change to the drive letter that corresponds to your CD-ROM drive, for example, *m:* (see [Figure 3.2](#)).

Figure 3.2: Using rawrite2 to make a boot disk

A screenshot of an MS-DOS Prompt window. The title bar reads "MS-DOS Prompt". The command prompt shows the following sequence of commands and output:

```
C:\WINDOWS>m:
M:\>tools\rawrite2\rawrite2
Rawrite 2.0 - Write disk file to raw floppy diskette

Enter disk image source file name: boot\rescl440.bin
Enter target diskette drive: a:
Please insert a formatted diskette into drive A: and press -ENTER- :
Number of sectors per track for this disk is 18
Writing image to drive A:. Press ^C to abort.
Track: 79 Head: 1
Done.

M:\>
```

4. In the MS-DOS window, type the command `cd tools\rawrite2\rawrite2` and press **Enter**.
5. When prompted, specify the file name of the disk image source as *boot\rescl440.bin* and press **Enter**.
6. When prompted, specify the drive letter of your floppy drive, for example, *a:*.
7. As instructed by the program, place a formatted floppy diskette in your floppy drive and press **Enter**.

It takes perhaps a minute or so for the `rawrite2` utility to create the floppy diskette. Wait for the utility to complete and then restart your system using the floppy diskette.

3.1.2.3 Starting the installation procedure

When Linux boots, you should see the `boot:` prompt shown in [Figure 3.3](#). Press **Enter** to begin the installation process.

Figure 3.3: The boot prompt

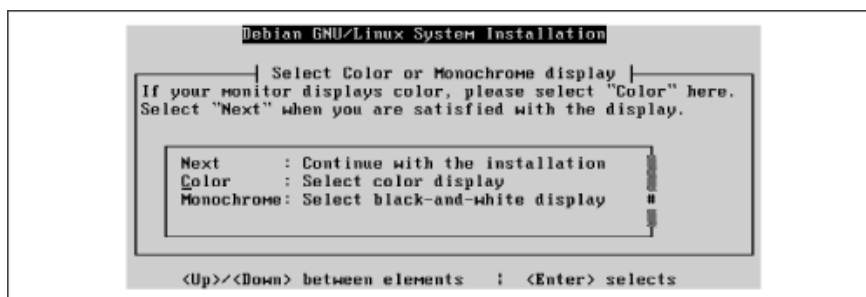


The `boot:` prompt lets you enter various kernel options. Most systems can be started without using any kernel options. However, if you cannot successfully boot your system from a CD-ROM or floppy diskette, you should suspect that a kernel option is needed. Pressing **F1** in response to the boot prompt will access some help pages. If the information in the help pages is not sufficient to resolve your problem, seek help as described in the section titled [Section 3.2, "Getting Help"](#), near the end of this chapter.

3.1.2.4 Choosing color versus monochrome

Once the install program starts, it first displays the Select Color or Monochrome Display screen, shown in [Figure 3.4](#), which asks whether subsequent install program screens should appear in color or monochrome (black and white). Use the Up and Down keys to move to the type of monitor attached to your system and press **Enter** to select it. If you selected Color, the screen reappears in color. To move forward to the next screen, highlight Next and press **Enter**.

Figure 3.4: The Select Color or Monochrome Display screen



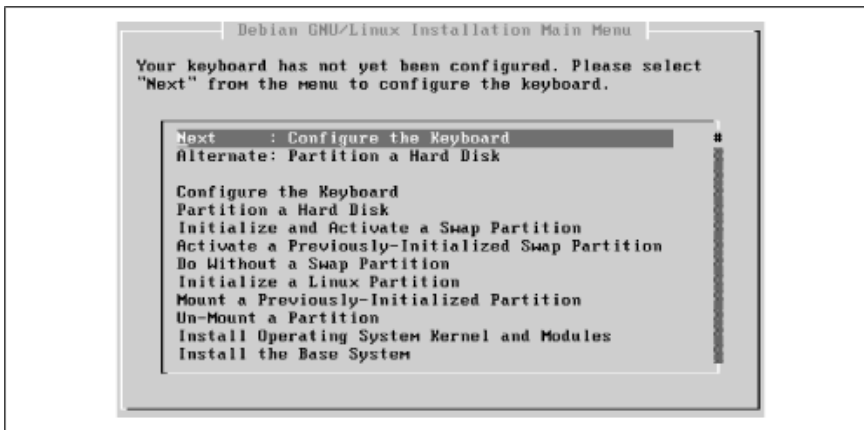
3.1.2.5 Release notes

The install program displays the current release notes. These identify the diskette used to boot the system, point you to the Debian web site, and explain the Debian mission. Use the Up and Down arrow keys to read text beyond the first page. When you've read the release notes, highlight Continue and press **Enter**.

3.1.2.6 The Installation Main Menu

The install program now displays the Installation Main Menu, shown in [Figure 3.5](#). This menu guides you through the installation process; it reappears in slightly different form after each installation step is completed.

Figure 3.5: The Installation Main Menu



The only control on the menu is a scrollable list. The first few items of the list present the installation steps that you should most likely perform next. The most likely step is labeled Next.

The remaining items of the list present other installation steps. If an installation goes awry, you can manually select the proper sequence of steps to quickly get things back on track.

However, you'll seldom, if ever, need this capability; choosing Next is almost always the appropriate action. To continue by configuring your system's keyboard, highlight Next and press **Enter**.

3.1.2.7 Selecting a keyboard

The install program displays the Select a Keyboard screen, shown in [Figure 3.6](#). Use the Up and Down arrow keys to select the appropriate keyboard. Most U.S.

users will prefer the pre-selected choice, U.S. English (QWERTY). When you've selected the proper keyboard, press ENTER to continue.

Figure 3.6: The Select a Keyboard screen



The installation main menu re-appears, with the Next choice designated Partition a Hard Disk. Press **Enter** to continue.

3.1.2.8 Selecting the hard drive

The Select Disk Drive screen, shown in [Figure 3.7](#), appears. The screen contains a scrollable list that lets you choose the drive to be partitioned. Drives are named by using the standard Linux method. IDE hard drives are named `hd x`, where `x` is a letter from `a` to `z`. Drive `hda` is your system's first IDE hard drive, drive `hdb` is your system's second IDE hard drive, and so on. SCSI drives are named `scd x`, where `x` is a letter from `a` to `z` that corresponds to the SCSI drive's disk ID number. As explained on the screen, the install program may mistakenly identify a CD-ROM drive as a hard drive.

Figure 3.7: The Select Disk Drive screen

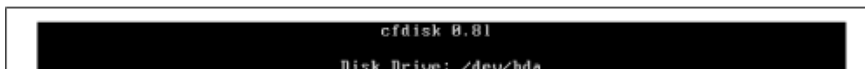


Choose the drive that you want to partition and press **Enter**.

3.1.2.9 Partitioning a hard drive

The install program launches `cfdisk`, a Linux program for partitioning hard drives. [Figure 3.8](#) shows the initial `cfdisk` screen. The screen shows the partitions and free space residing on the hard disk. In the figure, the hard disk contains only free space.

Figure 3.8: The initial cfdisk screen



To create a new partition from the available free space, use the Up and Down arrow keys to select a free space entry. Then use the Left and Right arrow keys to select the New menu item at the bottom of the screen. Press **Enter** to create the partition.

As shown in [Figure 3.9](#), `cfdisk` asks whether the new partition should be a primary or logical partition. Choose Primary and press **Enter**.

A hard disk can have a maximum of four primary partitions; a logical partition lets you escape this limitation. After creating a logical partition, you can create several extended partitions within it. However, `cfdisk` is not able to create extended partitions. If your hard disk already contains several partitions, you'll need to seek help in using a program other than `cfdisk` to partition your hard disk. See the section titled [Section 3.2](#)," near the end of this chapter.

Figure 3.9: Specifying the partition type



Next, `cfdisk` asks you to specify the size (in MB) of the new partition. As a rule of thumb, you should leave 50-100 MB of free space in which to establish a

Linux swap partition. Type the desired size, which must not exceed the available free space, and press **Enter**.

Next, if the new partition is smaller than the available free space, you're asked whether the new partition should be created at the beginning or the end of the available space. It generally makes little difference. Select Beginning or End, according to your preference, and press **Enter**.

An updated display appears, as shown in [Figure 3.10](#). The updated display shows the new partition, which was created as a Linux partition. Next, you must specify the Linux partition as bootable. Select the Bootable menu item by using the **Left** and **Right** arrow keys and press **Enter**. The screen is updated to reflect the new status of the partition. Notice how the new partition is named by using the name of the hard disk (`hda`) and a sequential number (1). Make a note of the name of the Linux partition.

Figure 3.10: The updated display

 Figure 3.10

Now, you must create a Linux swap partition from the remaining free space. Use the Up and Down arrow keys to select the free space and use the Left and Right arrow keys to select the New menu item. Press **Enter**.

Create the swap partition as a primary partition, with a size of 50-100 MB. Make a note of the name of the swap partition, which will be something like `hda2`.

Next, you must identify the new partition as a swap partition. Use the **Left** and **Right** arrow keys to select the Type menu item and press **Enter**. Type the code that corresponds to a Linux swap partition (82) and press **Enter**.

Finally, you must write the modified partition table to the hard disk. Use the **Left** and **Right** arrow keys to select the Write menu item and press **Enter**. The program tells you that erroneous changes to the partition table can destroy data. Check your work and, if the partition information is correct, type Yes and press **Enter**.

If the partition information is not correct, you can easily revise it. Select the erroneous partition and use the Delete menu item to delete the partition. You can then use the New menu item to recreate the partition with the proper size and type.

The program confirms that the partition table was written by displaying a message near the bottom of the screen. Exit the program by using the **Left** and **Right** arrow keys to select the Quit menu item and pressing **Enter**.

The main menu appears, with the Next item designated Initialize and Activate a Swap Partition. Press **Enter** to proceed.

3.1.2.10 Initializing and activating a swap partition

The install program asks you to identify the swap partition. Using your notes, select the proper partition - the smaller of the two partitions you just created - and press **Enter**.

The install program asks if you want the partition scanned for bad blocks. For a small partition such as a swap partition, this takes only a few moments and can help you avoid puzzling problems. Select Yes and press **Enter**.

The install program then informs you that all data on the swap partition will be destroyed. Make certain that you've correctly identified the partition, select Yes, and press **Enter** to begin the initialization. The display helps you keep track of the progress of the task.

When initialization is complete, the main menu reappears, with the Next item designated Initialize a Linux Partition. Press **Enter** to proceed.

3.1.2.11 Initializing a Linux partition

The install program asks you to identify the Linux partition. Using your notes, select the proper partition - the larger of the two partitions you earlier created - and press **Enter**.

The install program asks if you want the partition scanned for bad blocks. For a large partition, this can take several minutes. However, identifying and marking bad blocks can help you avoid puzzling problems, particularly if your hard disk hasn't been previously used. Select Yes and press **Enter**.

The install program then informs you that all data on the Linux partition will be destroyed. Make certain that you've correctly identified the partition, select Yes, and press **Enter** to begin the initialization. The display helps you keep track of the progress of the task.

When initialization is complete, the install program asks whether the Linux partition should be mounted as the root file system, the one to which programs will be installed. Select Yes and press **Enter** to mount the partition.

When the partition has been mounted, the main menu reappears, with the Next item designated Install Operating System Kernel and Modules. Press **ENTER** to proceed.

3.1.2.12 Installing the Operating System Kernel and Modules

The install program asks you to specify the medium which contains the Linux distribution. Use the Up and Down arrow keys to select CDROM and press **Enter**.

As shown in [Figure 3.11](#), the install program asks you to identify the CD-ROM drive that contains the distribution. Highlight the appropriate device and press **Enter**. If you can't confidently identify the device, don't fret. If the install program fails to find the distribution, you'll get another chance to identify the device.

Figure 3.11: Selecting the CD interface type

 Figure 3.11

The install program prompts you to place the distribution CD-ROM in the CD-ROM drive. Do so and then press **Enter**.

The install program prompts you to specify the CD-ROM directory that contains the distribution files. The text box is initialized with the default directory */debian*, which is the appropriate choice for the CD-ROM that accompanies this book. Simply use the Up and Down arrow keys to highlight Ok and press **Enter**.

The install program next asks how you want to specify the location of the *resc1440.bin* file that contains the kernel. Select the item designated List and press **Enter**.

The install program builds a list that contains the name of each directory that contains a file named *resc1440.bin*. The CD-ROM that accompanies this book includes only one such directory, so you can simply press **Enter** to select that directory.

The install program copies the kernel and modules to the hard drive. Then the main menu reappears, with the Next item designated Configure Device Driver Modules. Press **Enter** to proceed.

3.1.2.13 Configuring device driver modules


The install program prompts you to select a module category, by presenting the screen shown in [Figure 3.12](#). Each category contains a list of modules, small programs that extend the capability of the kernel to accommodate special devices and functions. By using the Select Category screen and its subscreens, you can specify which modules should be automatically loaded when you boot your Linux system.

Figure 3.12: Selecting a module category

 Figure 3.12

To see how this works, select the item designated "fs" and press **Enter**. The screen shown as [Figure 3.13](#) appears. As the screen explains, you can specify that a module should be loaded by selecting the corresponding list item, and pressing **Enter**. Select the item designated "binfmt_aout" and press **Enter**.

Figure 3.13: Selecting fs modules

 Figure 3.13

A confirmation screen, shown in [Figure 3.14](#) appears. To install the module, select the Install item and press **Enter**. A text-mode screen appears briefly to display the progress and result of installing the module. When the module has been installed, press **Enter** to return to the module selection screen.

Figure 3.14: Installing a module

 Figure 3.14

[Table 3.3](#) shows the modules you should install. If your computer has a network adapter, you should also install the proper module from the net category. If your computer has a CD-ROM with a proprietary interface (one other than ATAPI or SCSI), you should also install the proper module from the cdrom category. If you fail to install the proper modules, you can easily install them later. However, the

device corresponding to a module will not function until the module has been installed.

When installing some modules, such as the `lp` module, the install program may prompt you for additional information, using a screen similar to that shown in [Figure 3.15](#). Most modules will load and operate correctly even if you specify no command-line arguments. So, the simplest approach is to forego them. If a device associated with a module fails to operate correctly, you should suspect that command-line arguments are needed. Use the information in the [Section 3.2](#)" section to learn what arguments to specify and how to specify them.

Table 3.3: Modules To
Install

Category	Module
fs	binfmt_aout
fs	smbfs
fs	vfat
misc	lp
misc	serial
net	bsd_comp
net	dummy
net	ppp

Figure 3.15: Entering command-line arguments

 Figure 3.15

When you've specified all the necessary modules, exit the Select Category screen by highlighting Exit and pressing **Enter**. Then the main menu reappears, with the Next item designated Configure the Network. Press **Enter** to proceed.

3.1.2.14 Configuring the network

The install program presents a screen, shown in [Figure 3.16](#) that lets you choose a hostname for your system. If your system is attached to a network, the network administrator has likely assigned a hostname; in that case, you should specify that hostname here. Otherwise, you may select a hostname of your own choosing. Simply type the hostname in the text box, use the Up and Down arrow keys to highlight the Ok button, and press **Enter**.

Figure 3.16: Choosing the hostname

 Figure 3.16

The install program asks whether your computer is connected to a network. If your computer is permanently connected to a network, highlight Yes and press **Enter**. Otherwise, even if your computer temporarily connects to a network via a dialup connection, highlight No and press **Enter**.

If you specified that your computer is connected to a network, the install program presents a series of screens that prompt you for information describing your system and the network to which it connects. Your network administrator should provide you with the information required by the install program, including:

- Domain Name, the domain name of your system (for example, *debian.org*).
- IP Address, the network address of your system (for example, 192.168.1.2).
- Netmask, a bitmask that specifies the portion of your system's network address that uniquely identifies the network (for example, 255.255.255.0).
- Broadcast Address, which specifies the network address to which broadcast messages will be sent.
- Gateway, the network address of the router your system uses to send packets beyond its local network (for example, 192.168.1.1).

- Nameservers, the network addresses of the systems that provide hostname lookup services to your system (for example, 192.168.1.1).
- Type of primary network interface (for example, Ethernet or token ring).

The first such screen prompts you to specify the domain name associated with the network. Domain names are often - though not always - two words separated by a dot: for example, *oreilly.com*. Type the domain name associated with your network, highlight the Ok button, and press **Enter**. The install program asks you to verify the full name of your computer, which consists of the hostname and the domain name. Check your work. If the full name is correct, highlight Yes and press **Enter**. Otherwise, highlight No and press **Enter**; doing so will allow you to re-specify the erroneous information.

The install program next asks you to specify the IP address of your computer, which usually consists of four one- to three-digit numbers, separated by dots: for example, 192.168.1.1. Type the IP address, highlight the Ok button, and press **Enter**.

Next, the install program asks you to specify the netmask, which has a form resembling that of the IP address. Generally, the netmask value includes only the numbers 0 and 255. Type the netmask, highlight the Ok button, and press **Enter**. If you don't know the netmask, you can try the value 255.255.255.0, which is often correct.

Next, the install program asks you to specify your system's broadcast address. Generally, the first item - which specifies that the broadcast address is formed by setting the last several bits of the IP address to one - is the correct choice. Highlight the desired item and press **Enter**.

Next, the install program asks whether your computer connects via a gateway to networks other than its local network. If a gateway exists, highlight Yes and press **Enter**; otherwise highlight No and press **Enter**.

If you specified that a gateway exists, the install program prompts you for the IP address of the gateway system. Type the IP address, highlight the Ok button, and press **Enter**.

Next, the install program asks about nameservers. Generally, another system acts as the nameserver for a desktop system; therefore, unless your network administrator suggests otherwise, select the first item and press **Enter**.

If you specified that your system uses a nameserver, the install program asks for the nameserver's IP address. You can actually specify as many as five nameservers so that if one server is unavailable, your system can use another. Type one or more IP address, separating each address from the next by a space. Then, highlight the Ok button and press **Enter**.

The install program presents a confirmation screen, like that shown in [Figure 3.17](#), which summarizes your system's network configuration. If the configuration information is correct, highlight Yes and press **Enter** to proceed. Otherwise, highlight No and press **Enter** in order to be able to correct the erroneous information.

Figure 3.17: Confirming the network configuration

 Figure 3.17

When you've confirmed the network configuration, the install program asks you to specify the type of the primary network interface, by presenting the screen shown in [Figure 3.18](#). Generally, systems are connected to their network via an Ethernet card. Unless your network administrator suggests differently, select the "eth0" entry and press **Enter**.

Figure 3.18: Choosing the network interface

 Figure 3.18

After you select the network interface, the main menu reappears, with the Next item designated Install the Base System. Press **Enter** to proceed.

3.1.2.15 Installing the base system

The sequence of screens that follows resembles the sequence that appeared earlier when you installed the operating system kernel and modules. The install program first asks what medium contains the Linux distribution; select cdrom and press **Enter**.

Next, the install program asks you to identify the CD-ROM drive that contains the distribution. Highlight the appropriate device and press **Enter**. If you can't confidently identify the device, don't fret. If the install program fails to find the distribution, you'll get another chance to identify the device.

The install program prompts you to place the distribution CD-ROM in the CD-ROM drive. Do so and then press **Enter**.

The install program prompts you to specify the CD-ROM directory that contains the distribution files. The text box is initialized with the default directory */debian*, which is the appropriate choice for the CD-ROM that accompanies this book. Simply use the Up and Down arrow keys to highlight **Ok** and press **Enter**.

The install program next asks how you want to specify the location of the *base2_1.tgz* file that contains the first part of the base system. Select the item designated **List** and press **Enter**.

The install program builds a list that contains the name of each directory that contains a file named *base2_1.tgz*. The CD-ROM that accompanies this book includes only one such directory, so you can simply press **Enter** to select that directory.

The install program copies the base system to the hard drive. Then the main menu reappears, with the Next item designated **Configure the Base System**. Press **Enter** to proceed.

3.1.2.16 Configuring the base system

To configure the base system, you must first select the time zone associated with the system's location. You can do this either of two ways, by using the screen shown in [Figure 3.19](#). The list at the left of the screen (titled *Timezones*) lets you select a time zone by its coded designation. The list at the right of the screen (cryptically titled *Directories*) lets you select a time zone by location, using familiar geographical names. Use the Left and Right arrow keys to select the list you want to use, then use the Up and Down arrow keys to select the appropriate list item and press **Enter**. If you selected a location, the install program may present a second screen that lets you more precisely designate the location. Use the Up and Down arrow keys to select the desired item and press **Enter**.

Figure 3.19: Selecting the Time Zone

 Figure 3.19

Next, the install program presents a screen that explains that Unix system clocks are generally set to Universal Time (GMT), rather than local time. However, you're free to set your system clock any way you like. Most desktop users prefer to set their system's clock to local time, particularly if their system contains

multiple operating systems. Press **Enter** to move forward to the screen that lets you specify how you want your system's clock to be interpreted. If you plan to set your system's clock to GMT, select Yes and press **Enter**; otherwise, select No and press **Enter**.

The main menu reappears, with the Next item designated Make Linux Bootable Directly from Hard Disk.

Do not press **Enter**. Instead, use the Up and Down arrow keys to select the Alternate item, designated Make a Boot Floppy. By booting Linux from a floppy, you avoid several potential problems. For further information on booting Linux, including information on booting Linux directly from a hard disk, see **Appendix D, *Managing the Boot Process***.

3.1.2.17 Making a boot floppy

Next, the install program instructs you to place a blank floppy diskette in your system's first floppy disk drive. Insert a floppy diskette - it need not be formatted - and press **Enter**. The install program creates the boot floppy.

WARNING: All data on the floppy disk will be lost.

After the floppy diskette has been created, the main menu appears, with the Next item designated Reboot the System. Press **Enter**.

The install program asks you to confirm your decision to reboot the system. Leave the newly created boot floppy in the floppy disk drive, highlight Yes, and press **Enter**. Your system should restart. If - after a minute or so - it hasn't restarted, press the system's reset button or cycle power to the system.

A boot prompt should appear shortly after your system completes its power-on self test. Press **Enter** to boot Linux from the floppy diskette. Linux should load, causing a series of messages to cascade off the screen.

3.1.3 Configuring the New System

Once your system has booted, you'll see a series of screens that prompt you to configure your new system. First, you're prompted to establish a password for the `root` user.

3.1.3.1 Establishing a root password

When your system has booted, you should see the screen shown in [Figure 3.20](#). This screen lets you establish a password for the `root` user, the master user who administers the system. Follow the on-screen instructions, by typing a password consisting of from five to eight characters. If you choose a password that the install program regards as insecure, the system will prompt you to type another password. If you really want the original password, simply type it again; the system will not object a second time. For security, the system will not display a password as it's typed. Instead, it will prompt you to enter the password a second time, helping you avoid typing a password other than the one you intend.

Figure 3.20: Setting a password for root

 Figure 3.20

3.1.3.2 Establishing a normal user account

Next, the install program asks whether you want to create a normal user account, in addition to the `root` user account. Respond by typing `Y` and pressing **Enter**. Then type a username, consisting of eight characters or less and including only letters and digits. Many Linux users create usernames that consist of the first letter of their first name followed by their last name, or the first seven characters of their last name if their last name has eight or more characters. For example, the author often uses the username `bmccarty`. After typing the username, press **Enter**. If you make a mistake, use the backspace key to erase the erroneous letters.

The system will ask you to establish a password for the new user account, much as it did for the `root` account. It then asks you to supply the following additional information:

Full Name

Your full name, including your first and last name

Room Number

The room number of your office

Work Phone

Your work phone number

Home Phone

Your home phone number

Other

Other brief information about the user

You should supply the full name. You can supply or omit the other information as you see fit. Once you've entered all the information, the system asks you to confirm your entries. Type Y and press **Enter** to confirm your choices; or, type N and press **Enter** to revise your choices.

3.1.3.3 Configuring shadow passwords

Next, the system asks whether to configure shadow passwords. Unless you have a good reason for doing otherwise, you should configure shadow passwords by typing Y and pressing **Enter**. That way, your system's encrypted passwords are stored in a file that only the `root` user can read, making your system much less susceptible to break-ins.

3.1.3.4 Removing PCMCIA support

If your system doesn't require PCMCIA support, the system prompts you to allow it to remove the PCMCIA modules. So long as you don't actually require PCMCIA support, you should type Y and press **Enter**.

3.1.3.5 Connecting via PPP

At this point, depending on options you previously selected, the system may prompt you to allow it to connect via PPP to download requested packages. If this prompt appears, respond by typing N and pressing **Enter**. Downloading the packages over a dial-up connection would be a very time-consuming operation; it's much quicker to access the packages from the CD-ROM that accompanies this book.

3.1.3.6 Choosing Packages

Next, the system asks whether you'd like to use a speedy way of specifying what applications (packages) you want to install. Type Y and press **Enter** to take the fast route, which lets you select from a set of system profiles or tasks.

The screen shown in [Figure 3.21](#) appears. This screen lets you choose profiles that describe the kind of system you want, or tasks that describe the kinds of operations you want your system to perform. The system associates a set of applications with each profile or task; selecting a profile or task instructs the system to install the associated applications in an upcoming step.

Figure 3.21: Selecting a profile

Figure 3.21

The most appropriate profile for most initial users of Linux is Basic. To choose that profile, use the Up and Down arrow keys to select the Basic item. Then select the Ok button and press **Enter**.

Next, the system informs you that it's about to start the `dselect` program, which actually installs the selected applications. Press **Enter** to begin.

3.1.4 Installing the Applications

The initial `dselect` screen, the program's main menu, is shown in [Figure 3.22](#). You can use the Up and Down arrow keys to highlight a menu item. You can select the highlighted menu item by pressing **Enter**. The on-screen instructions earlier advised that you should skip Access and Update: *ignore these instructions*.

Instead, you'll generally select the menu items in order in which they appear. However, you'll skip menu item 2 (Select), menu item 4 (Config), and menu item 5 (Remove).

Figure 3.22: The dselect main menu

Figure 3.22

3.1.4.1 Accessing the packages

Highlight the Access menu item and press **Enter**. The program presents a list of methods for accessing the applications to be installed. Use the Up and Down arrow keys to highlight the entry designated apt and press **Enter**.

Next, you'll be asked if you want to change the source list. Respond *Yes*, which initiates a dialog that builds a simple configuration. Here's a sample dialog that

shows the responses you should give to install packages from the CD-ROM diskette that accompanies this book:

```
I see you already have a source list.
```

```
-----
```

```
source list displayed here: contents vary
```

```
-----
```

```
Do you wish to change it?[y/N]
```

y

```
Set up a list of distribution source locations
```

```
Please give the base URL of the debian distribution.
```

```
The access schemes I know about are: http ftp file
```

```
For example:
```

```
file:/mnt/debian,  
ftp://ftp.debian.org/debian,  
http://ftp.de.debian.org/debian,
```

```
URL [http://http.us.debian.org/debian]:
```

file:/cdrom

```
Please give the distribution tag to get or a path to the  
package file ending in a /. The distribution  
tags are typically something like: stable unstable frozen non-US
```

```
Distribution [stable]:
```

stable

```
Please give the components to get
```

```
The components are typically something like: main contrib non-free
```

```
Components [main contrib non-free]:
```

main contrib

```
Would you like to add another source?[y/N]
```

N

3.1.4.2 Mounting the CD-ROM

Access the second virtual console by pressing **Alt-F2**. Login as `root` and issue the following commands:

```
mkdir /cdrom  
mount -t iso9660 -o ro /dev/
```


`hdx /cdrom`

where `hdx` represents the Linux designation of the CD-ROM drive that contains the diskette that accompanies this book. For example, `hdb` is the secondary drive on the first controller. See [Figure 3.1](#) for other common designations.

If the command succeeds, return to the first virtual console by pressing **Alt-F1**. Otherwise, check the arguments and try again.

3.1.4.3 Updating the list of available packages

Highlight the Update menu item and press **Enter**. The program will update its list of available packages. When prompted to do so, press **Enter** to return to the main menu.

3.1.4.4 Installing the selected packages

Skip the Select menu item, by highlighting the Install menu item. Press **Enter** to begin installing the selected packages.

A mail package called `exim` will request configuration information as it's installed. [Table 3.4](#) summarizes these requests and provides the information with which you should respond.

Table 3.4: Configuration Information for `exim`

Prompt	Response
Select a number from 1 to 5....	2
Visible mail name of your system	Enter
Other system names appearing on incoming names	Enter

Prompt	Response
Domains for which mail is relayed	Enter
Local machines for which mail is relayed	Enter
RBL (sites from which mail will not be accepted)	Enter
Smart host handling outgoing mail	The outgoing mail host provided by your Internet service provider, if any
User account for system administration mail	Enter

You can revise the configuration of `exim` after installation by issuing the command `eximconfig`.

When all the selected packages have been installed, the program will display the message "Installation OK." Press **Enter** to return to the main menu.

3.1.4.5 Exiting dselect

Exit the `dselect` program by highlighting the Quit menu item and pressing **Enter**. The screen shown in [Figure 3.23](#) appears.

Figure 3.23: The Linux login prompt

 Figure 3.23

To login to Linux for the first time, type *root* or the name of the user account you earlier created, and then press **Enter**. When the system prompts for the password, type the password and press **Enter**. You should then see a shell

prompt, consisting of a pound sign (#) or dollar sign (\$), as shown in [Figure 3.24](#). Congratulations: you've just installed and configured Linux.

Figure 3.24: The Linux shell prompt

 Figure 3.24

If you don't see a shell prompt, or if something went wrong earlier in the installation process, don't despair. The next section will show you how to obtain the help you need to get your Linux system up and running.

3.2 Getting Help

If your system fails to boot, or if you're unable to complete the Linux installation process, don't despair. The help you need is probably close by, in one of these sources:

- The Debian Web Site
- Linux FAQs (Frequently Asked Questions)
- Linux HOWTOs
- The Debian mailing lists
- USENET newsgroups
- Internet Relay Chat (IRC)

The following sections describe these sources and explain how to access and use them. You should generally consult them in the order specified.

3.2.1 The Debian Web Site

The Debian Web site, <http://www.debian.org/>, is your main source for information on Debian GNU/Linux. It provides documents, links, and resources galore.

3.2.2 FAQs

Linux FAQs present commonly asked questions and answers. The Debian GNU/Linux FAQ is available online at <http://www.debian.org/doc/FAQ/>. You can also find it on the CD-ROM that accompanies this book, in the `/doc/FAQ` directory. The Linux FAQ is available at <http://metalab.unc.edu/LDP/FAQ/Linux-FAQ-1.html>.

3.2.3 HOWTOs

Linux HOWTOs address specific topics of interest to Linux users. They're found in the `/usr/doc/HOWTO` directory of an installed Debian GNU/Linux system. You can also find them online at <http://metalab.unc.edu/linux/intro.html>. At this point in your Linux experience, you'll probably find the Installation-HOWTO

useful. Use it to find workarounds for your installation and configuration problems.

The HOWTOs are generally available in plain text format. You can use Microsoft Windows WordPad, or another text editor of your choice, to access them.

3.2.4 The Debian Mailing Lists

Several Internet mailing lists address Debian GNU/Linux. You can find a list of these at <http://www.debian.org/MailingLists/subscribe/>. Among the most useful mailing lists are these:

debian-user

The main mailing list for Debian GNU/Linux users and developers.

debian-laptop

Issues related to installing, updating and using Debian GNU/Linux on laptop computers.

debian-boot

Issues related to booting Debian GNU/Linux.

debian-announce

Important announcements directed to the Debian GNU/Linux community.

debian-security-announce

Issues related to security, including fixes to security problems.

debian-commercial

Information on commercial products related to Debian GNU/Linux.

debian-news

General news about Debian GNU/Linux and the Debian project.

debian-changes

Changes to Debian GNU/Linux releases are published here.

3.2.5 USENET Newsgroups

Several Internet newsgroups address Linux. You can find a list of these at <http://metalab.unc.edu/linux/intro.html>. If your Internet Service Provider (ISP) provides access to these newsgroups - as most do - you can read and post messages read by other Linux users around the world. If necessary, consult your ISP for information on accessing these newsgroups.

Don't post blindly to these newsgroups, or you may draw angry responses; instead, you should first attempt to find answers to your questions in the Linux FAQs and HOWTOs. Generally, the Linux community is quite willing to help even those who ask what some consider dumb questions; but, as a courtesy to all, it's best if you do some work on your own before seeking the help of others.

3.2.6 Internet Relay Chat (IRC)

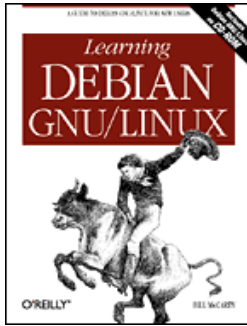
IRC (Internet Relay Chat) lets you exchange typed messages in real time with people from all over the world. The server irc.debian.org provides an IRC channel dedicated to Debian GNU/Linux. To connect to the server, you must have an installed IRC client such as BitchX, ircII, tkirc, or Zircon. If you're having trouble getting Debian GNU/Linux installed or configured, you can use a Windows-based IRC client to chat with Debian users, who can likely help you resolve your problem.

Connect to the server by typing:

```
/server irc.debian.org
```

Then join channel *#debian* by typing:

```
/join #debian
```



Learning Debian GNU/Linux

By [Bill McCarty](#)

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

4. Issuing Linux Commands

Contents:

[The System Use Cycle](#)

[Working with the Linux Command Prompt](#)

[How Linux Organizes Data](#)

[Working with Devices](#)

[Useful Linux Programs](#)

This chapter shows you how to begin using your Linux system. It shows you how to boot your system, log in, issue commands, log out, and shut down your system. It also explains how to use the `man` command, which provides help on using other commands. The chapter describes how Linux organizes data as filesystems, directories, and files and how you can work with removable media, such as diskettes. It describes how to query the status of your system. And, finally, it explains how to use `pico`, a simple text editor.

4.1 The System Use Cycle

This section introduces you to the cycle of Linux system use. If you're a user of Microsoft Windows, you're accustomed to a pattern of system use that forms a cycle:

- Boot the system
- Identify yourself to the system
- Use the system
- Shutdown the system

The cycle of Linux system use is similar, even though you perform the tasks somewhat differently.

4.1.1 Booting the System

Most Linux users boot their system from its hard drive. Of course, if you made a boot diskette during system installation, you can use it to boot your system.

First, you must prepare your system for booting. If your system is running, you must shut it down by following the proper procedure for shutting down the operating system that's active. For example, if you're running Microsoft Windows, click Start → Shut Down and select the Shut Down option in the Shut Down dialog box. Press OK to begin the system shutdown. After a few seconds, Windows displays a screen telling you that it's safe to turn off power to your system. Turn off the power or, if your system automatically powers down, wait a few seconds until the system powers itself down.

Next, you must set your system to boot from the desired device. To boot your system from its hard drive, remove any floppy diskette from your system's floppy drive. To boot your system from a floppy diskette, insert your Linux boot diskette into your system's floppy drive.

Now, you're ready to boot your system. Switch your system on (or press your system's reset button, if your system is powered on) and watch as it performs its self test. Shortly thereafter, you should see a `boot:` prompt on the system's monitor. If you like, you can list the available boot configurations stored on the boot device by pressing **Tab**. To boot the system, type the name of the desired configuration and press **Enter**, or simply press **Enter** to boot using the default configuration.

Once it loads, Linux begins probing your system and its devices, printing status information on your system's monitor. This status information is helpful if your system fails to boot properly, because it discloses the point in the boot process where the problem occurred.

When Linux has completed its boot process, your system's monitor will display a login prompt similar to this:

```
Debian GNU/Linux 2.1 desktop tty1
desktop login:
```

4.1.2 Logging In

Before you can use the system, you must identify yourself by logging in. The install program created a special user named `root`; by identifying yourself as the `root` user, you can gain access to the system. Normally, you use the `root` userid only when performing system administration tasks, because the `root` user has special capabilities that other users lack.

To log on, type `root` and press **Enter**. The system prompts you for the password associated with the `root` userid. Type the password you established during the installation process and press **Enter**. To prevent anyone nearby from learning your password, Linux does not display it as you type. If you suspect you've typed it incorrectly, simply press **Enter** and start over; or press **Backspace** once (or more) for each character you've entered and then re-enter it. If you type the userid or password incorrectly, Linux displays the message "login incorrect" and prompts you to try again.

Like other members of the Unix family, the Linux operating system is case sensitive. Be sure to type the userid `root` just as it appears, using all lowercase characters. Similarly, you must type the password exactly as you entered it in the Root Password dialog box during system installation.

Also, some Linux programs require you to type Ctrl-BACKSPACE, rather than BACKSPACE. If you press BACKSPACE and see `^H` echoed to the console, try pressing Ctrl-BACKSPACE instead.

When you've successfully logged in, you'll see a command prompt that looks something like this:

```
root@desktop:/root#
```

This prompt tells you that the Linux `bash` shell is ready to accept your commands.

4.1.3 Issuing Commands

The component of Linux that interprets and executes commands is called the *shell*. Linux supports a variety of different shells, but the most popular is the `bash` shell. This chapter presents the basics of using the `bash` shell; you'll learn more about the shell in **Chapter 13**,

Conquering the BASH Shell.

The Linux `bash` shell presents the user with a command-line interface (CLI). CLIs are familiar to Windows users who have worked in the MS-DOS Prompt window, and indeed the Microsoft Windows MS-DOS Prompt window is a kind of command-line shell for Windows. The Linux `bash` shell works much like the MS-DOS Prompt window. You type text commands and the system responds by displaying text replies. As your first Linux command, type `w` and press **Enter**. Your screen should look something like this:

```
root@desktop:/root#
```

w

```
11:12am  up 6 min,   1 user,   load average: 0.00, 0.08, 0.05
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT
root      tty1                11:13am  0.00s   0.20s   0.11s  -bash
```

The `w` command tells Linux to display the system status and a list of all system users. In the example, the output of the command tells you that it's now 11:12 a.m., that the system has been up for 6 minutes, and that only one user - `root` - is currently logged in. Notice that the command output is very terse, packing much information into a few lines. Such output is typical of Linux commands. At first, you may find Linux output cryptic and difficult to read, but over time you'll grow to appreciate the efficiency with which Linux communicates information.

Linux provides many commands besides the `w` command; so many that you may despair of learning and recalling them. Actually, the number of commands you'll use regularly is fairly small. Soon, these will become second nature to you.

Now try a second command, the `date` command:

```
root@desktop:/root#
```

date

```
Tue Feb 23 11:15:20 PST 1999
```

The `date` command displays the current date and time.

If you find working with MS-DOS distasteful or intimidating, you may not immediately enjoy working with the Linux command line. However, give yourself some time to adjust. The Linux command line has several features that make it easier to use, and more powerful, than MS-DOS. If, after working with the Linux command line for several days, you don't find yourself at home, don't despair. Linux provides a graphical user interface in addition to its command-line interface. You'll learn about the graphical user interface in **Chapter 6**,

Using the X Window System.

4.1.4 Correcting Commands

Sometimes you may type a command incorrectly, causing Linux to display an error message. For example, suppose you typed `dat` instead of `date`:

```
root@desktop:/root#
```

```
dat
```

```
bash: dat: command not found
```

In such a case, carefully check the spelling of the command and try again. If you notice an error before pressing **Enter**, you can use the **Backspace** key to return to the point of the error and then type the correct characters.

Just as a web browser keeps track of recently visited sites, Linux's BASH shell keeps track of recently issued commands. This list is called the history list, and you can scroll back through it using the Up arrow key, or back down using the Down arrow key, just as you would with the Back and Forward buttons on a web browser. In fact, the history list provides several powerful ways to remember and reuse frequently issued commands, as we'll see in **Chapter 13**.

The Up and Down arrow keys let you scroll through a list of commands recently issued. This feature is handy when you want to repeat a command. Simply use the Up arrow key to find the command and press **Enter** to re-execute it. You can also use this feature when you want to issue a command similar to one you recently issued. Use the Up arrow key to find the original command. Then, use the Left and Right arrow keys to position the cursor and make whatever changes to the command you like. Finally, press **Enter** to execute the command.

4.1.5 Using Virtual Consoles

In Microsoft Windows, you can have several MS-DOS Prompt windows simultaneously active. Although the `bash` shell doesn't have a graphical user interface, you can nevertheless work with several instances of the shell, by using Linux virtual consoles. Linux provides six virtual consoles; you can use special keystrokes to switch between them. The keystroke **Alt-F *n***, where *n* is the number of a virtual console (1-6), causes Linux to display virtual console *n*. For example, you can display virtual console 2 by typing **Alt-F2**. You can view only a single console at a time, but you can switch rapidly between consoles by using the appropriate keystroke.

Virtual consoles are handy when you've started a time-consuming task and want to be able to perform an unrelated task while the original task is working. You'll also

find them useful after you've established several userids on your system, because you can log on as one userid on one virtual console while you're logged on as another userid on a different console.

Virtual consoles have a screen saver feature like that found on Microsoft Windows. If a virtual console is inactive for an extended period, Linux blanks the monitor screen. To restore the screen without disturbing its contents, press the **Shift** key.

4.1.6 Logging Out

When you're done using a virtual console, you should log out by typing the command `exit` and pressing **Enter**. When you log out, the system frees memory and other resources that were allocated when you logged in, making those resources available to other users.

When the system logs you out, it immediately displays a login prompt. If you change your mind and want to access the system, you can login simply by supplying your userid and password.

4.1.7 Shutting Down the System

You shouldn't turn off power to a computer while it's running Linux; instead, you should shut down the operating system and then turn off power. To shut down a Linux system, you use the `shutdown` command, which resides in a directory named */sbin*:

```
root@desktop:/root#  
/sbin/shutdown -h now
```

Don't type the prompt, which automatically appears on the command line. Only the `root` user can issue the `shutdown` command. If you want to restart a Linux system, you can use an alternative form of the `shutdown` command:

```
root@desktop:/root#  
/sbin/shutdown -r now
```

Or, even more conveniently, you can use the familiar MS-DOS "three-finger salute": **Ctrl-Alt-Del**, which simply issues a `shutdown` command on your behalf.

When you shut down a system, Linux automatically logs off all users, terminates all running programs, and closes all open files. Before shutting down a system, you should check each virtual console to determine if an important operation is in progress. If so, you should delay shutting the system down until the operation completes.

4.2 Working with the Linux Command Prompt

To make Linux commands easy to use, they share a simple, common structure. This section describes their common structure and explains how you can obtain helpful information on the commands available to you.

4.2.1 Command Structure

Linux commands share the common form:

```
command option(s) argument(s)
```

The *command* identifies the command you want Linux to execute. The name of a Linux command almost always consists of lowercase letters and digits. Remember that, unlike Microsoft Windows, Linux is case sensitive; be sure to type each character of a command in the proper case.

Most commands let you specify options or arguments. However, in any given case, you may not need to do so. For example, typing the `w` command without options and arguments causes Linux to display a list of current users.

Options modify the way that a command works. Most options consist of a single letter, prefixed by a dash. Often, you can specify more than one option; when you do so, you separate each option with a space or tab. For example, the `-h` option of the `w` command causes the output of the command to omit the header lines that give the time and the names of the fields. Typing:

```
root@desktop:/root#
```

```
w -h
```

prints a list of users without the header lines.

Arguments specify filenames or other targets that direct the action of the command. For example, the `w` command lets you specify a userid as an argument, which causes the command to list only logins that pertain to the specified userid. Typing:

```
root@desktop:/root#
```

```
w root
```

prints a list of current logins by the `root` user. Some commands let you specify a series of arguments; you must separate each argument with a space or tab.

4.2.2 Getting Help

Because Linux provides so many commands and because Linux commands provide so many possible options, you can't expect to recall all of them. To help you, Linux provides the `man` command and the `apropos` command, which let you access a help database that describes each command and its options.

4.2.2.1 Using `man`

Each Linux command is described by a special file called a *manual page*. The manual pages are stored in a group of subdirectories comprising a help database. To access this database, you use the `man` command, which resembles the MS-DOS `help` command. For example, to get help on using the `w` command, type:

```
root@desktop:/root#
```

```
man w
```

[Figure 4.1](#) shows the resulting output, which the command displays one page at a time. Notice the colon prompt, which appears at the bottom left of the screen. To page forward, press the **Space** key; to page backward, press the **b** key. To exit the `man` program, press the **q** key.


The manual pages are organized according to a common format. At the beginning of a manual page, you'll find the name of the page and the section of the manual page database from which the page comes, shown in parentheses. For example, the figure shows the manual page named `w`, which comes from section 1 of the manual page database. [Table 4.1](#) describes the sections of the manual page database; most sections are primarily of interest to programmers. As a user and administrator, you'll be interested primarily in sections 1 and 8.

Table 4.1: Manual Page Sections

Section	Description
1	Executable programs and shell commands
2	System calls (provided by the kernel)

Section	Description
3	Library calls (provided by system libraries)
4	Special files (for example, device files)
5	File formats and conventions
6	Games
7	Macro packages and conventions
8	System administration commands
9	Non-standard kernel routines

Figure 4.1: A typical man page

 Figure 4.1

Next in the output comes the name and brief description of the command. Then comes a synopsis of the command, which shows the options and arguments that you can specify. Brackets enclose parts of a command that you can choose to include or omit. Next comes a detailed description of the operation of the command, followed by a description of its options.

As you're learning your way around Linux, you may find it convenient to reserve a virtual console for running the `man` command. That way, you can enter commands in a separate virtual console, switching between consoles to refresh your recollection of the options and arguments of commands as you type them.

4.2.2.2 Using apropos

The `man` command searches the manual pages and displays detailed information about a specified command. The `apropos` command also searches the manual pages; however, it displays summary information about manual pages that contain a specified keyword. (The search is limited to the short description that appears at the beginning of each manual page). For example, typing the command:

```
root@desktop:/root#
```

```
apropos files
```

displays a list of manual pages that contain the word *files*, as shown in [Figure 4.2](#).

Figure 4.2: Output of the apropos command

 Figure 4.2

The `apropos` command is useful when you don't recall the name of a Linux command. By typing a related keyword, you can obtain a list of commands and search the list for the command you need.

4.3 How Linux Organizes Data

In order to make the most effective use of your Linux system, you must understand how Linux organizes data. If you're familiar with Microsoft Windows or another operating system, you'll find it easy to learn how Linux organizes data, because most operating systems organize data in rather similar ways. This section explains how Linux organizes data. It also introduces you to several important Linux commands that work with directories and files.

4.3.1 Devices

Linux receives data from, sends data to, and stores data on *devices*. A device usually corresponds to a hardware unit, such as a keyboard or serial port. However, a device may have no hardware counterpart: the kernel creates several *pseudodevices* that you can access as devices but that have no physical existence. Moreover, a single hardware unit may correspond to several devices - for example, Linux defines each partition of a disk drive as a distinct device. [Table 4.2](#) describes some typical Linux devices; not every system provides all these devices and some systems provide devices not shown in the table.

Table 4.2: Typical Linux Devices

Device	Description
<code>atibm</code>	Bus mouse
<code>audio</code>	Sound card
<code>cdrom</code>	CD-ROM drive
<code>console</code>	Current virtual console
<code>fd <i>n</i></code>	Floppy drive (<i>n</i> designates the drive; for example, <code>fd0</code> is the first floppy drive)

Device	Description
<code>ftape</code>	Streaming tape drive not supporting rewind
<code>hd xn</code>	Non-SCSI hard drive (x designates the drive and n designates the partition; for example, <code>hda1</code> is the first partition of the first non-SCSI hard drive)
<code>inportbm</code>	Bus mouse
<code>lp n</code>	Parallel port (n designates the device number; for example, <code>lp0</code> is the first parallel port)
<code>modem</code>	Modem
<code>mouse</code>	Mouse
<code>nftape</code>	Streaming tape drive supporting rewind
<code>nrft n</code>	Streaming tape drive supporting rewind (n designates the device number; for example, <code>nrft0</code> is the first streaming tape drive)
<code>nst n</code>	Streaming SCSI tape drive not supporting rewind (n designates the device number; for example, <code>nst0</code> is the first streaming SCSI tape drive)
<code>null</code>	Pseudodevice that accepts unlimited output
<code>printer</code>	Printer

Device	Description
<code>psaux</code>	Auxiliary pointing device, such as a trackball, or the knob on IBM's Thinkpad
<code>rft <i>n</i></code>	Streaming tape drive not supporting rewind (<i>n</i> designates the device number; for example, <code>rft0</code> is the first streaming tape drive)
<code>scd <i>n</i></code>	SCSI device (<i>n</i> designates the device number; for example, <code>scd0</code> is the first SCSI device)
<code>sd <i>xn</i></code>	SCSI hard drive (<i>x</i> designates the drive and <i>n</i> designates the partition; for example, <code>sda1</code> is the first partition of the first SCSI hard drive)
<code>sr <i>n</i></code>	SCSI CD-ROM (<i>n</i> designates the drive; for example, <code>sr0</code> is the first SCSI CD-ROM)
<code>st <i>n</i></code>	Streaming SCSI tape drive supporting rewind (<i>n</i> designates the device number; for example, <code>st0</code> is the first streaming SCSI tape drive)
<code>tty <i>n</i></code>	Virtual console (<i>n</i> designates the particular virtual console; for example, <code>tty0</code> is the first virtual console)
<code>ttyS <i>n</i></code>	Modem (<i>n</i> designates the port; for example, <code>ttyS0</code> is an incoming modem connection on the first serial port)
<code>zero</code>	Pseudodevice that supplies an inexhaustible stream of zero-bytes

4.3.2 Filesystems

Whether you're using Microsoft Windows or Linux, you must format a partition before you can store data on it. When you format a partition, Linux writes special data, called a *filesystem*, on the partition. The filesystem organizes the available

space and provides a directory that lets you assign a name to each *file*, which is a set of stored data. You can also group files into *directories*, which function much like the folders you create using the Microsoft Windows Explorer: directories store information about the files they contain.

Every CD-ROM and floppy diskette must also have a filesystem. The filesystem of a CD-ROM is written when the disk is created; the filesystem of a floppy diskette is rewritten each time you format it.

Microsoft Windows 95 lets you choose to format a partition as a FAT or FAT32. Linux supports a wider variety of filesystem types; [Table 4.3](#) summarizes the most common ones. The most important filesystem types are `ext2`, which is used for Linux native partitions, `msdos`, which is used for FAT partitions (and floppy diskettes) of the sort created by MS-DOS and Microsoft Windows; and `iso9660`, which is used for CD-ROMs. Linux also provides the `vfat` filesystem, which is used for FAT32 partitions of the sort created by Microsoft Windows 9x. Linux also supports reading of Windows NT NTFS filesystems; however, the support for writing such partitions is not yet stable.

Table 4.3: Common Filesystem Types

Filesystem	Description
coherent	A filesystem compatible with that used by Coherent Unix
ext	The predecessor of the <code>ext2</code> filesystem; supported for compatibility
ext2	The standard Linux filesystem
hpfs	A filesystem compatible with that used by IBM's OS/2
iso9660	The standard filesystem used on CD-ROMs
minix	An old Linux filesystem, still occasionally used on floppy diskettes

Filesystem	Description
msdos	A filesystem compatible with Microsoft's FAT filesystem, used by MS-DOS and Windows
nfs	A filesystem compatible with Sun's Network File System
ntfs	A filesystem compatible with that used by Microsoft Windows NT's NTFS filesystem
sysv	A filesystem compatible with that used by AT&T's System V Unix
vfat	A filesystem compatible with Microsoft's FAT32 filesystem, used by Windows 9x
xenix	A filesystem compatible with that used by Xenix

4.3.3 Directories

If you've used MS-DOS, you're familiar with the concepts of file and directory, and with various MS-DOS commands that work with files and directories. Under Linux, files and directories work much as they do under MS-DOS.

4.3.3.1 Home and working directories

When you login to Linux, you're placed in a special directory known as your *home directory*. Generally, each user has a distinct home directory, where the user creates personal files. This makes it simple for the user to find files previously created, because they're kept separate from the files of other users.

The *working directory* - or *current working directory*, as it's sometimes called - is the directory you're currently working in. When you login to Linux, your home directory is your working directory. By using the `cd` command (which you'll meet in a moment) you can change your working directory.

4.3.3.2 The directory tree

The directories of a Linux system are organized as a hierarchy. Unlike MS-DOS, which provides a separate hierarchy for each partition, Linux provides a single hierarchy that includes every partition. The topmost directory of the directory tree is the *root directory*, which is written using a forward slash (/), not the backward slash (\) used by MS-DOS to designate a root directory.

[Figure 4.3](#) shows a hypothetical Linux directory tree. The root directory contains six subdirectories: *bin*, *dev*, *etc*, *home*, *tmp*, and *usr*. The *home* directory has two subdirectories; each is the home directory of a user and has the same name as the user who owns it. The user named *bill* has created two subdirectories in his home directory: *books* and *school*. The user named *patrick* has created a single subdirectory in his home directory: *school*.

Figure 4.3: A hypothetical Linux directory tree

Figure 4.3

Each directory (other than the root directory) is contained in a directory known as its parent directory. For example, the parent directory of the *bill* directory is *home*.

4.3.3.3 Absolute and relative pathnames

Notice in the figure that two directories named *school* exist: One is a subdirectory of *bill* and the other is a subdirectory of *patrick*. To avoid confusion that could result when several directories have the same name, directories are specified using *pathnames*. Two kinds of pathnames exist: *absolute* and *relative*. The absolute pathname of a directory traces the location of the directory beginning at the root directory; you form the pathname as a list of directories, separated by forward slashes (/). For example, the absolute pathname of the unique directory named *bill* is */home/bill*. The absolute pathname of the *school* subdirectory of the *bill* directory is */home/bill/school*. The absolute pathname of the identically named *school* subdirectory of the *patrick* directory is */home/patrick/school*.

When a subdirectory is many levels below the root directory, its absolute pathname may be long and cumbersome. In such a case, it may be more convenient to use a relative path name, which uses the current directory, rather than the root directory, as its starting point. For example, suppose that the *bill* directory is the current working directory; you can refer to its *books* subdirectory by the relative pathname *books*. Notice that a relative pathname can never begin with a forward slash, whereas an absolute pathname must begin with a forward slash. As a second example, suppose that the *home* directory is the current working directory. The

relative pathname of the *school* subdirectory of the *bill* directory would be *bill/school*; the relative pathname of the identically named subdirectory of the *patrick* directory would be *patrick/school*.

Linux provides two special directory names. Using a single dot (.) as a directory name is equivalent to specifying the working directory. Using two dots (..) within a pathname takes you up one level in the current path, to the parent directory. For example, if the working directory is */home/bill*, *..* refers to the */home* directory. Similarly, the path *../patrick/school* refers to the directory */home/patrick/school*.

4.3.4 Commands That Work with Directories

Now that you understand the fundamentals of how Linux organizes data, you're ready to learn some commands that work with directories. Rather than simply read this section, you should login to your Linux system and try the commands for yourself. Only by doing so will you begin to develop skill in working with shell commands.

4.3.4.1 Displaying the working directory

To display the current working directory, issue the `pwd` command. The `pwd` command requires no options or arguments.

```
root@desktop:/root#
```

```
pwd  
/root
```

The `pwd` command displays the absolute pathname of the working directory.

4.3.4.2 Changing the working directory

To change the working directory, issue the `cd` command, specifying the pathname of the new working directory as an argument. You can use an absolute or relative pathname. For example, to change the working directory to the */bin* directory, type:

```
root@desktop:/root#  
  
cd /bin  
[root@desktop /bin]#
```

Notice how the prompt changes to indicate that */bin* is now the working directory.

You can quickly return to your home directory by issuing the `cd` command without an argument:

```
[root@desktop /bin]#
```

```
cd
```

```
root@desktop:/root#
```

Again, notice how the prompt changes to indicate the new working directory.

If you attempt to change the working directory to a directory that doesn't exist, Linux displays an error message:

```
root@desktop:/root#
```

```
cd nowhere
```

```
bash: nowhere: No such file or directory
```

4.3.4.3 Displaying directory contents

To display the contents of a directory, you use the `ls` command. The `ls` command provides many useful options that let you tailor its operation and output to your liking.

The simplest form of the `ls` command takes no options or arguments. It simply lists the contents of the working directory, including files and subdirectories (your own output will differ, reflecting the files present in your working directory):

```
root@desktop:/root#
```

```
ls
```

```
GNUstep          firewall          sniff
Xrootenv.0       linux            ssh-1.2.26
audio.cddeb      mail            ssh-1.2.26.tar.gz
audio.wav        mirror          support
axhome          mirror-2.8.tar.gz temp
conf            nlxb3181.tar    test
corel           openn           test.doc
drivec.img      scan            tulip.c
dynip_2.00.tar.gz screen-3.7.6-0.i386.rpm win95
root@desktop:/root#
```

Here, the output is presented in lexical (dictionary) order, as three columns of data. Notice that filenames beginning with uppercase letters appear before those beginning with lowercase letters.

A more sophisticated form of the `ls` command that includes the `-l` option displays descriptive information along with the filenames, as shown in [Figure 4.4](#).

Figure 4.4: Output of the `ls` command

 Figure 4.4

The first line of the output shows the amount of disk space used by the working directory and its subdirectories, measured in 1K blocks. Each remaining line describes a single file or directory. The columns are:

Type

The type of file: a directory (`d`), or an ordinary file (`-`). If your system supports color, Linux displays output lines that pertain to directories in blue and lines that pertain to files in white.

Access modes

The access mode, which determines the users that can and cannot access the file or directory.

Links

The number of files or directories linked to this one.

Group

The group that owns the file or directory.

Size

The size of the file or directory, in bytes.

Modification date

The date and time when the file or directory was last modified.

Name

The name of the file or directory.

You'll learn more about access modes, links, and groups in subsequent sections of this chapter.

If a directory contains many files, the listing will fill more than one screen. To view the output one screen at a time, use the command:

```
ls -l | more
```

This command employs the pipe redirector (`|`, explained in **Chapter 13**), sending output of the `ls` subcommand to the `more` subcommand, which presents the output

one screen at a time. You can control the operation of the `more` command by using the following keys:

- **Space** moves you one page forward
- `b` moves you one page back
- `q` exits the program and returns you to the command prompt

If you want to list a directory other than the working directory, you can type the name of the directory as an argument of the `ls` command. Linux displays the contents of the directory, but does not change the working directory. Similarly, you can display information about a file by typing its name as an argument of the `ls` command. Moreover, the `ls` command accepts indefinitely many arguments, so you can type a series of directories and filenames as arguments, separating each with one or more spaces or tabs.

When the name of a directory or file begins with a dot (`.`), the output of the `ls` command does not normally include the directory or file, which is said to be *hidden*. To cause the output of the `ls` command to include hidden directories and files, use the `-a` option. For example, to list all the files and subdirectories in the current directory - including hidden ones - type:

```
root@desktop:/root#  
  
ls -a -l
```

If you prefer, you can combine the `-a` and `-l` options, typing the command like this:

```
root@desktop:/root#  
  
ls -al
```

A user's home directory generally includes several hidden files containing configuration information for various programs. For example, the `.profile` file contains configuration information for the Linux shell.

The `ls` command provides a host of additional useful options; see its manual page for details.

4.3.4.4 Creating a directory

You can create directories by using the `mkdir` command. Just type the name of the new directory as an argument of the command. Linux creates the directory as a subdirectory of the working directory. For example, this command creates a subdirectory named *office*:

```
root@desktop:/root#
```

```
mkdir office
```

If you don't want to create the new directory as a subdirectory of the working directory, type an absolute or relative pathname as the argument. For example, to create a directory named */root/documents*, type:

```
root@desktop:/root#
```

```
mkdir /root/documents
```

This works regardless of the current working directory.

The name of a directory or file must follow certain rules. For example, it must not contain a slash (/) character. Directory and file names usually include letters (either uppercase or lowercase), digits, dots, and underscores (_). You can use other characters, such as spaces, but such names present problems, because the shell gives them special meaning. If you simply must use a name containing special characters, enclose the name within single quotes (' '). The quotes don't become part of the name that is stored on the disk. This technique is useful when accessing files on a Microsoft Windows filesystem; otherwise you'll have trouble working with files in directories such as *My Documents*, which have names containing spaces.

Most MS-DOS filenames contain a dot, but most Linux filenames do not. In MS-DOS, the dot separates the main part of the filename from a part known as the extension, which denotes the type of the file. For example, the MS-DOS file *memo.txt* would contain text. Most Linux programs determine the type of a file automatically, so Linux filenames don't require an extension.

4.3.4.5 Removing a directory

To remove a directory, use the `rmdir` command. For example, to remove *unwanted*, a subdirectory of the working directory, type:

```
root@desktop:/root#
```

```
rmdir unwanted
```

If the directory you want to delete is not a subdirectory of the working directory, remove it by typing an absolute or relative pathname.

You cannot remove a directory that contains files or subdirectories with `rmdir`; you must first delete the files in the directory and then remove the directory itself.

However, the command `rm -r` will recursively remove the files in a directory and then remove the directory.

4.3.5 Working with Files

Directories contain files and other directories. You use files to store data. This section introduces you to several useful commands for working with files.

4.3.5.1 Displaying the contents of a file

Linux files, like Microsoft Windows files, can contain text or binary information. The contents of a binary file are meaningful only to skilled programmers, but you can easily view the contents of a text file. Simply type the `cat` command, specifying the name of the text file as an argument. For example:

```
root@desktop:/root#
```

```
cat /etc/passwd
```

displays the contents of the `/etc/passwd` file, which lists the valid system logons.

If a file is too large to be displayed on a single screen, the first part of the file will whiz past you and you'll see only the last few lines of the file. To avoid this, you can use the `more` command:

```
root@desktop:/root#
```

```
more /etc/passwd
```

This command displays the contents of a file in the same way the `man` command displays a manual page. You can use **Space** and the **b** key to page forward and backward through the file and the **q** key to exit the command.

4.3.5.2 Removing a file

To remove a file, type the `rm` command, specifying the name of the file as an argument. For example:

```
root@desktop:/root#
```

```
rm badfile
```

removes the file named *badfile* contained in the working directory. If a file is located elsewhere, you can remove it by specifying an absolute or relative pathname.

WARNING: Once you remove a Linux file, its contents are lost forever. Be careful to avoid removing a file that contains needed information.

The `-i` option causes the `rm` command to prompt you to verify your decision to remove a file. If you don't trust your typing skills, you may find this option helpful. Linux automatically supplies the `-i` option even if you don't type it.

4.3.5.3 Copying a file

To copy a file, use the `cp` command, specifying the name (or path) of the file you want to copy and the name (or path) to which you want to copy it. For example:

```
root@desktop:/root#
```

```
cp /etc/passwd sample
```

copies the `/etc/passwd` file to a file named *sample* in the working directory.

If the destination file already exists, Linux overwrites it. You must therefore be careful to avoid overwriting a file that contains needed data. Before copying a file, use the `ls` command to ensure that no file will be overwritten; alternatively, use the `-i` option of the `cp` command, which prompts you to verify the overwriting of an existing file. Linux automatically supplies the `-i` option even if you don't type it.

4.3.5.4 Renaming or moving a file

To rename a file, use the `mv` command, specifying the name (or path) of the file and the new name (or path). For example:

```
root@desktop:/root#
```

```
mv old new
```

renames the file named *old* as *new*. If the destination file already exists, Linux overwrites it, so you must be careful. Before moving a file, use the `ls` command to ensure that no file will be overwritten; or, use the `-i` option of the `mv` command, which prompts you to verify the overwriting of an existing file. Linux automatically supplies the `-i` option even if you don't type it.

The `mv` command can rename a directory, but cannot move a directory from one device to another. To move a directory to a new device, first copy the directory and its contents and then remove the original.

4.3.5.5 Finding a file

If you know the name of a file, but do not know what directory contains it, you can use the `find` command to locate the file. For example:

```
root@desktop:/root#
```

```
find . -name 'missing' -print
```

attempts to find a file named *missing*, located in (or beneath) the current working directory (.). If the command finds the file, it displays its absolute pathname.

If you know only part of the file name, you can surround the part you know with asterisks (*):

```
root@desktop:/root#
```

```
find / -name '*iss*' -print
```

This command will find any file whose name includes the characters *iss*, searching every subdirectory of the root directory (that is, the entire system).

4.3.5.6 Printing a file

If your system includes a printer, you can print a file by using the `lpr` command. For example:

```
root@desktop:/root#
```

```
lpr /etc/passwd
```

sends the file */etc/passwd* to the printer.

If a file is lengthy, it may require some time to print. You can send other files to the printer while a file is printing. The `lpq` command lets you see what files are queued to be printed:

```
root@desktop:/root#
```

```
lpq
```

```
lp is ready and printing
```

Rank	Owner	Job	Files	Total Size
active	root	155	/etc/passwd	1030 bytes

Each waiting or active file has an assigned print job number. You can use the `lprm` to cancel printing of a file, by specifying the print job number. For example:

```
root@desktop:/root#
```

```
lprm 155
```


cancels printing of job number 155. However, only the user who requested that a file be printed (or the root user) can cancel printing of the file.

4.3.5.7 Working with compressed files

To save disk space and expedite downloads, you can compress a data file. By convention, compressed files are named ending in `.gz`; however, Linux doesn't require or enforce this convention.

To expand a compressed file, use the `gunzip` command. For example, suppose the file *bigfile.gz* has been compressed. Typing the command:

```
root@desktop:/root#
```

```
gunzip bigfile.gz
```

extracts the file *bigfile* and removes the file *bigfile.gz*.

To compress a file, use the `gzip` command. For example, to compress the file *bigfile*, type the command:

```
root@desktop:/root#
```

```
gzip bigfile
```

The command creates the file *bigfile.gz* and removes the file *bigfile*.

Sometimes it's convenient to store several files (or the contents of several subdirectories) in a single file. This is useful, for example, in creating a backup or archive copy of files. The Linux `tar` command creates a single file that contains data from several files. Unlike the `gzip` command, the `tar` command doesn't disturb the original files. To create a *tar file*, as a file created by the `tar` command is called, a command like this:

```
tar -cvf  
tarfile files-or-directories
```

Substitute *tarfile* with the name of the tar file you want to create and *files-or-directories* with a list of files and directories, separating the list elements by one or more spaces or tabs. You can use absolute or relative pathnames to specify the files or directories. By convention, the name of a tar file ends with *.tar*, but Linux does not require or enforce this convention.

For example, to create a tar file named *backup.tar* that contains all the files in all subdirectories of the directory */home/bill*, type:

```
tar -cvf backup.tar /home/bill
```

The command creates the file *backup.tar* in the current working directory.

You can list the contents of a tar file by using a command that follows this pattern:

```
tar -tvf
tarfile | more
```

The `| more` causes the output to be sent to the `more` command, so that you can page through multiple pages. If the tar file holds only a few files, you can omit the `| more`.

To extract the contents of a tar file, use a command that follows this pattern:

```
tar -xvf
tarfile
```

This command expands the files and directories contained within the tar file as files and subdirectories of the working directory. If a file or subdirectory already exists, it is silently overwritten.

The `tar` command provides a host of useful options; see its manual page for details.

It's common to compress a tar file. You can easily accomplish this by specifying the options `-czvf` instead of `-cvf`. Compressed tar files are conventionally named ending with *.tgz*. To expand a compressed tar file, specify the options `-xzvf` instead of `-xvf`.

The `tar` command doesn't use the common ZIP method of compression common in the Microsoft Windows world. However, Linux can easily work with, or even create, ZIP files. To create a ZIP file that holds compressed files or directories, issue a command like this one:

```
zip -r
zipfile files_to_zip
```

where *zipfile* names the ZIP file that will be created and *files_to_zip* specifies the files and directories to be included in the ZIP file.

To expand an existing ZIP file, issue a command like this one:

```
unzip
zipfile
```

4.3.5.8 Working with links

Microsoft Windows 9x supports shortcuts, which let you refer to a file or directory (folder) by several names. Shortcuts also let you include a file in several directories or a subdirectory within multiple parent directories. In Linux, you accomplish these results by using the `ln` command, which links multiple names to a single file or directory. These names are called *symbolic links*, *soft links*, or simply *links*.

To link a new name to an existing file or directory, type a command that follows this pattern:

```
ln -s  
old new
```

For example, suppose that the current working directory contains the file *william*. To be able to refer to this same file by the alternative name *bill*, type the command:

```
root@desktop:/root#  
  
ln -s william bill
```

The `ls` command shows the result:

```
root@desktop:/root#  
  
ls -l  
lrwxrwxrwx    1 root      root              7 Feb 27 13:58 bill->william  
-rw-r--r--    1 root      root            1030 Feb 27 13:26 william
```

The new file (*bill*) has type `l`, which indicates it's a link, rather than a file or directory. Moreover, the `ls` command helpfully shows the name of the file to which the link refers (*william*).

If you omit the `-s` option, Linux creates what's called a *hard link*. A hard link must be stored on the same filesystem as the file to which it refers, a restriction that does not apply to symbolic links. The link count displayed by the `ls` command reflects only hard links; symbolic links are ignored.

4.3.5.9 Working with file permissions

Unlike Windows 98, but like other varieties of Unix and Windows NT, Linux is a multi-user operating system. Therefore, it includes mechanisms that protect data from unauthorized access. The primary protection mechanism restricts access to directories and files, based on the identity of the user who requests access and on *access modes* assigned to each directory and file, which are often called *permissions*.

Each directory and file has an associated user, called the *owner*, who created the directory or file. Each user belongs to one or more sets of users known as *groups*.

Each directory and file has an associated group, which is assigned when the directory or file is created.

Groups can be used to let users other than `root` perform system administration tasks or other restricted tasks. For example, Debian GNU/Linux defines the group `dialout`; only members of this group - and `root`, of course - can access the system's modem and initiate a dial-up connection. By allowing only the members of a particular group access to a program file, you can establish a flexible, yet effective security policy.

To restrict access to a file or directory, you set its permissions. [Table 4.4](#) lists the possible permissions and explains the meaning of each. Notice that permissions work differently for directories than for files. For example, permission *r* denotes the ability to list the contents of a directory or read the contents of a file. A directory or file can have more than one permission. Only the listed permissions are granted; any other operations are prohibited. For example, a user who had file permission *rw* could read or write the file, but could not execute it.

Table 4.4: Access Permissions

Permission	Meaning for directory	Meaning for file
r	List the directory	Read contents
w	Create or remove files	Write contents
x	Access files and subdirectories	Execute

The access modes of a directory or file consist of three permissions:

owner

Applies to the owner of the file

group

Applies to users who are members of the group assigned to the file

other

Applies to other users

The `ls` command lists the file access modes in the second column of its long output format, as shown in [Figure 4.5](#). The column contains nine characters: the first three specify the access allowed the owner of the directory or file, the second three specify the access allowed users in the same group as the directory or file, and the final three specify the access allowed to other users (see [Figure 4.6](#)).

Figure 4.5: Access modes as shown by the `ls` command

 Figure 4.5

Figure 4.6: Access modes specify three permissions

 Figure 4.6

You set the access modes of a directory or file by using the `chmod` command, which has the following pattern:

```
chmod
nnn directory-or-file
```

The argument *nnn* is a three-digit number, which gives the access mode for the owner, group, and other users. [Table 4.5](#) shows each possible digit and the equivalent access permission. For example, the argument 751 is equivalent to `rwxr-x--x`, which gives the owner every possible permission, gives the group read and execute permission, and gives other users execute permission.

Table 4.5:
Numerical
Access Mode
Values

Value	Meaning
0	---
1	--x

Value	Meaning
2	-w-
3	-wx
4	r--
5	r-x
6	rw-
7	rwX

If you're the owner of a file or directory (or if you're the root user), you can change its ownership by using the `chown` command. For example, the following command assigns *newuser* as the owner of the file *hotpotato*:

```
root@desktop:/root#
```

```
chown newuser hotpotato
```

The owner of a file or directory (and the root user) can also change the group of a file. For example, the following command assigns *newgroup* as the new group of the file *hotpotato*:

```
root@desktop:/root#
```

```
chgrp newgroup hotpotato
```

The group you assign to a file or directory must have been previously established by the root user. The valid groups appear in the file */etc/group*, which only the root user can alter.

The root user can assign each user to one or more groups. When you log on to the system, you are assigned to one of these groups - your *login group* - by default. To change to another of your assigned groups, you can use the `newgrp` command. For example, to change to the group named *secondgroup*, use the following command:

```
root@desktop:/root#
```

```
newgrp secondgroup
```

If you attempt to change to a group that does not exist, or to which you have not been assigned, your command will fail. When you create a file or directory, it is automatically assigned your current group as its owning group.

4.3.5.10 Running programs

In Linux, as in MS-DOS and Microsoft Windows, programs are stored in files. Often, you can launch a program by simply typing its filename. However, this assumes that the file is stored in one of a series of directories known as the *path*. A directory included in this series is said to be *on the path*. If you've worked with MS-DOS, you're familiar with the MS-DOS path, which works much like the Linux path. You'll learn more about working with the Linux path in **Chapter 13**.

If the file you want to launch is not stored in a directory on the path, you can simply type the absolute pathname of the file. Linux will then launch the program even though it's not on the path. If the file you want to launch is stored in the working directory, type `./` followed by the name of the program file. Again, Linux will launch the program even though it's not on the path.

For example, suppose the program *bigdeal* is stored in the directory */home/bob*, which is the current directory and which happens to be on the path. You could launch the program any of these ways:

```
bigdeal  
./bigdeal  
/home/bob/bigdeal
```

The first command assumes that the program is on the path. The second assumes that the program resides in the current working directory. The third makes no assumptions about the location of the file.

4.4 Working with Devices

This section presents commands that work with devices. You'll learn how to mount and unmount devices and how to format a floppy diskette.

4.4.1 Mounting and Unmounting Drives

You cannot access a hard drive partition, CD-ROM, or floppy diskette until the related device or partition is *mounted*. Mounting a device checks the status of the device and readies it for access. Linux can be configured to automatically mount a device or partition when it boots; but you must manually mount other devices and partitions.

If a device uses removable media, the media may not be present when the system boots. If the system is configured to automatically mount such a device and the media is not present, an error occurs. Therefore, devices that use removable media are not generally configured for automatic mounting.

Before you can remove media from a device, you must unmount it. The system also unmounts devices when it shuts down. Mounting and unmounting devices is a privileged operation; generally, only the root user can manually mount and unmount devices.

To mount a device or partition, you use the `mount` command, which has the following pattern:

```
mount  
options device directory
```

The `mount` command provides many options. However, you can generally use the `mount` command without any options; consult the manual page to learn about the available options.

The reason you can use the `mount` command without options is that the file `/etc/fstab` describes your system's devices and the type of filesystem each is likely to contain. If you add a new device to your system, you may need to revise the contents of `/etc/fstab` or specify appropriate options when you mount the device.

You must specify the device that you want to mount and a directory, known as the *mount point*. To make it convenient to access various devices, Linux treats a mounted device as a directory; mounting the device associates it with the named directory. For example, a common operation is mounting a CD-ROM. You can accomplish this with the command:


```
root@desktop:/root#
```

```
mount /dev/cdrom /cdrom
```

The file `/dev/cdrom` is a link that points to the actual device file associated with your system's CD-ROM drive. The directory `/cdrom` is a directory created by the install program; this directory is conventionally used as the mounting point for CD-ROMs. After the command has completed, you can access files and directories on the CD-ROM just as you would access ordinary files and directories on the path `/cdrom`. For example, to list the top-level files and directories of the CD-ROM simply type:

```
root@desktop:/root#
```

```
ls /cdrom
```

To mount a floppy diskette in your *a:* drive, type:

```
root@desktop:/root#
```

```
mount /dev/fd0 /floppy
```

To unmount a device, specify its mount point as an argument of the `umount` command. For example, to unmount a CD-ROM diskette, type:

```
root@desktop:/root#
```

```
umount /cdrom
```

Only the root user can unmount a device. Moreover, a device can be unmounted only if it's not in use. If, for example, the working directory of a user is a directory of the device, the device cannot be unmounted.

If you can't unmount a device, check each virtual console to see if one of them has a session that's using the device as its working directory. If so, either exit the session or change to a working directory that isn't associated with the device.

4.4.2 Formatting Floppy Diskettes

Before you can write data on a floppy diskette, you must format it. The Linux command to format a floppy is `fdformat`. Simply follow the command with an argument that specifies the floppy drive and the capacity of the diskette; the available arguments are listed in [Table 4.6](#). For example, to format a 1.44 MB floppy in your system's *a:* drive, type:

```
root@desktop:/root#
```

```
fdformat /dev/fd0H1440
```

Once you've formatted the floppy, you can mount it and then read and write it. Be sure you unmount the floppy diskette before you remove it. Unmounting the floppy diskette ensures that all pending data has been written to it; otherwise, the floppy diskette may be unusable due to corrupt data.

Table 4.6: Floppy Drive Designators

Designation	Meaning
<i>/dev/fd0</i>	3.5-inch diskette in <i>a</i> : (1.44 MB)
<i>/dev/fd0d360</i>	5.25-inch diskette in <i>a</i> : (360 kB)
<i>/dev/fd0D720</i>	3.5-inch diskette in <i>a</i> : (720 kB)
<i>/dev/fd0h1200</i>	5.25-inch diskette in <i>a</i> : (1.2 MB)
<i>/dev/fd0H1440</i>	3.5-inch diskette in <i>a</i> : (1.44 MB)
<i>/dev/fd0H2880</i>	3.5-inch diskette in <i>a</i> : (2.88 MB)
<i>/dev/fd1</i>	3.5-inch diskette in <i>b</i> : (1.44 MB)
<i>/dev/fd1d360</i>	5.25-inch diskette in <i>b</i> : (360 kB)
<i>/dev/fd1D720</i>	3.5-inch diskette in <i>b</i> : (720 kB)
<i>/dev/fd1h1200</i>	5.25-inch diskette in <i>b</i> : (1.2 MB)
<i>/dev/fd1H1440</i>	3.5-inch diskette in <i>b</i> : (1.44 MB)

Designation	Meaning
<i>/dev/fd1H2880</i>	3.5-inch diskette in <i>b</i> : (2.88 MB)

4.5 Useful Linux Programs

This section presents several programs you may find helpful in working with your Linux system. You'll learn several commands that report system status and you'll learn how to use `pico`, a simple text editor.

4.5.1 Viewing System Information

Linux provides a number of commands that report system status. The most commonly used commands are shown in [Table 4.7](#). These commands can help you troubleshoot system problems and identify resource bottlenecks. Although each command can be used without options or arguments, each supports options and arguments that let you customize operation and output; consult the appropriate manual page for details.

Table 4.7: Useful System Commands

Command	Function
<code>df</code>	Shows the amount of free disk space (in 1K blocks) on each mounted filesystem.
<code>du</code>	Shows the amount of disk space (in 1K blocks) used by the working directory and its subdirectories.
<code>free</code>	Shows memory usage statistics, including total free memory, memory used, physical memory, swap memory, shared memory, and buffers used by the kernel.
<code>ps</code>	Shows the active processes (instances of running programs) associated with this login session. Use the <code>-a</code> option to list all processes.
<code>top</code>	Shows a continually updated display of active processes, and the resources they are using. Type the <code>q</code> key to exit.

Command	Function
<code>uptime</code>	Shows the current time, the amount of time logged in, the number of users logged in, and system load averages.
<code>users</code>	Shows each login session.
<code>w</code>	Shows a summary of system usage, currently logged-in users, and active processes.
<code>who</code>	Shows the names of users currently logged in, the terminal each is using, the time each has been logged in, and the name of the host from which each logged in (if any).

4.5.2 Using the `ae` Editor

The `ae` editor is a simple text editor that you can think of as the Linux equivalent of the Microsoft Windows Notepad accessory. To start `ae`, simply type `ae` at the shell prompt; or, if you want to edit a particular file, type `ae` followed by the name of the file (or the file's path, if the file is not in the working directory). For example, to edit the file *mydata*, type:

```
root@desktop:/root#
```

```
ae mydata
```

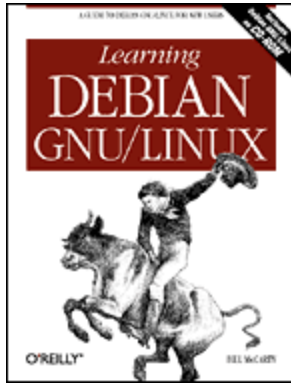
[Figure 4.7](#) shows `ae`'s standard display. At the top of the display is a status line, which shows the name of the file being edited and the current size of the file. If the file has been modified, the status line includes the word *Modified*. The top half of the display lists the available editing commands. Most of the commands require you to type a control character, so that commands can be distinguished from characters you want to add to the buffer. For example, `^x` denotes a Ctrl+X. To save the current file, you type a Ctrl+X followed by a Ctrl+S. Typing an ordinary character inserts it at the current cursor position. You can use the cursor keys to move around the display; you can use the delete or backspace key to erase unwanted characters. [Table 4.8](#) summarizes `ae`'s most useful commands

Figure 4.7: The ae editor

 Figure 4.7

Table 4.8: Summary of ae Commands

Command	Description
Ctrl+X I	Read a file into the editor.
Ctrl+X Ctrl+S	Write the current file.
Ctrl+X Ctrl+C	Save current file and exit.
Ctrl+Q	Quit without saving current file.
Ctrl+L	Redraw the screen.



Learning Debian GNU/Linux

By Bill McCarty

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

5. Installing and Configuring the X Window System

Contents:

[What is X?](#)

[Installing X](#)

[Configuring X](#)

[Starting and Stopping X](#)

This chapter helps you install, configure, and use the X Window System (often known simply as X). Once X is up and running, you can choose how to start X. This chapter explains your options and also gives some tips on optimizing the performance of X.

5.1 What is X?

X is the standard graphical user interface for Linux. Like other graphical user interfaces such as Microsoft Windows and Mac OS, X lets you interact with programs by using a mouse (or other pointing device) to point and click, providing a simple means of communicating with your computer.

Originally implemented as a collaborative effort of Digital Equipment Corporation and Massachusetts Institute of Technology, X was first released in 1987. Subsequently, the X Consortium, Inc. became responsible for the continued development and publication of X.

Despite its age, X is a remarkable and very modern software system: a cross-platform, network-oriented graphical user interface. It runs on a wide variety of platforms, including essentially every variety of Unix. X Clients are available for use, for example, under Windows 3.x, 9x, and NT. The sophisticated networking capabilities of X let you run a program on one computer while viewing the graphical output on another computer, connected to the first via a network. With the advent of the Internet, which interconnected a sizable fraction of the computers on the planet, X achieved a new height of importance and power.

Most Linux users run XFree86, a freely available software system compatible with X. XFree86 was developed by the XFree86 software team, which began work in 1992. In 1994, The XFree86 Project assumed responsibility for ongoing research and development of XFree86.

5.2 Installing X

Getting a proper X Window System up and running used to be a real challenge on Linux, almost a rite of passage. Today, device drivers are available for a much wider array of hardware, and configuration tools to assist in the setup process have greatly improved. While still tricky at times - especially with unusual hardware - X setup and configuration is no longer the daunting process it once was, and should be relatively easy.

You'll go through two stages before you have X successfully running. The first stage involves installing the needed programs that enable X to run. These can be grouped into several categories:

- Basic XFree86 program
- X servers
- Window Managers
- Applications
- Fonts

This stage is very straightforward and can even be done as part of the basic installation process, if you select the relevant X packages during that step.

In the second stage you configure X to run properly on your system. This is a matter of identifying an X server compatible with your graphics card, and tuning the server for your graphics card. If you have a common card and all the documentation for it, this second stage will be relatively simple. Missing information makes the process harder, but not impossible.

As shown in **Appendix C, *The Debian Package Management Utilities***, X consists of many packages.

Once you've installed the necessary packages, you're ready to configure X.

WARNING: You should exercise due care while configuring X to run on your system. If you incorrectly or incompletely configure X, your system can be permanently damaged. In particular, if you configure your monitor for a refresh rate that exceeds its capacity, you can damage the monitor. Older fixed-frequency monitors are particularly susceptible to such damage. The author and publisher have taken pains to make this chapter clear and accurate, but their efforts don't ensure that the procedure presented in this chapter will work correctly with your hardware. Consequently, the author and publisher cannot be held responsible for damages resulting from a faulty installation or configuration of X.

If you have a card or monitor of unknown manufacture or model, and feel that you must guess, at least start with a narrow range of middle values, and gradually expand that range to see if you can find a value that works. Don't let a monitor that displays an unstable or garbled image run any longer than the time it takes you to cut power to the monitor.

5.3 Configuring X

When you install the *xserver-common* package, `xf86config` is automatically launched. However, you can launch the program any time you like. To do so, log in as `root` and type the command:

```
xf86config
```

[Figure 5.1](#) shows the beginning of the `xf86config` dialog. As you can see, `xf86config` is a text-mode program; it does not support use of the mouse and it presents its questions teletype-style, using black-and-white text.

In working with `xf86config`, you may find that your **Backspace** key doesn't work as expected. If so, use **Ctrl-Backspace** instead.

Figure 5.1: The beginning of the `xf86config` dialog

 Figure 5.1

Next, `xf86config` asks you to specify the type of mouse attached to your system, as shown in [Figure 5.2](#). Type the number associated with your choice, and press **Enter**. For non-mouse pointing devices found on many laptops, you should most likely select PS/2 Mouse.

Figure 5.2: Specifying the mouse type

 Figure 5.2

If you selected the Logitech MouseMan mouse, you should enable its third button by responding *y* to the question asking whether ChordMiddle should be enabled, as shown in [Figure 5.3](#).

Figure 5.3: Specifying the ChordMiddle option

 Figure 5.3

If your mouse has only two buttons, you should enable emulation of a three-button mouse by responding *y* to the question asking about `Emulate3Buttons`, as shown in [Figure 5.4](#). If you enable this option you can simultaneously press both the buttons of your mouse to emulate pressing the third button.

Figure 5.4: Specifying emulation of a three-button mouse

 Figure 5.4

Next, you must specify the device file associated with the mouse, as shown in [Figure 5.5](#). The install program should have associated your system's mouse with the device `/dev/mouse`, which is the default choice. Simply press **Enter** to continue.

Figure 5.5: Specifying the mouse device

 Figure 5.5

As described by the output shown in [Figure 5.6](#), X provides special support for using extended keyboards. If you use a special keyboard layout to support national characters, you can type *y* to use `xkb`, which simplifies changing the keyboard map. After making your choice, press **Enter** to continue.

Figure 5.6: Specifying use of the keyboard extension

 Figure 5.6

As shown in [Figure 5.7](#), `xkb` supports a variety of keyboard encodings, or keymaps. Type the number that corresponds to the type of keyboard attached to your system, and press **Enter**.

Figure 5.7: Specifying the keymap

 Figure 5.7

Next, as shown in [Figure 5.8](#), you must specify two characteristics of your system's monitor: its vertical refresh rate (VertRefresh) and horizontal sync rate (HorizSync). You can find these values by:

- Consulting your monitor's documentation
- Consulting the file `/usr/doc/xserver-common/Monitors.gz`, which may list your monitor. Use `gunzip` to uncompress the file and `ae` or another text editor to view it.
- Viewing the monitor's manufacturer's web support page
- Posting a question to the newsgroup [comp.os.linux.setup](#)
- Contacting the monitor manufacturer's technical support group and requesting the information

To specify the monitor's characteristics, press **Enter**.

Figure 5.8: Preparing to specify monitor characteristics

 Figure 5.8

First, you must specify the horizontal sync rate of your monitor, as shown in [Figure 5.9](#). Type the number associated with your choice and press **Enter**. If you're unsure of your monitor's horizontal sync range, but certain that it supports 800×600 resolution, specify range 2. To specify a range other than those listed, you can select choice 11; if you do so, you'll be prompted to enter the low and high values of the horizontal sync range.

WARNING: Often, otherwise similar monitor models have different horizontal sync rates. It is crucial that you accurately determine the horizontal sync rate of your monitor. If you configure X to use an inappropriate horizontal sync rate, you can permanently damage your monitor.

Figure 5.9: Specifying the horizontal sync rate

 Figure 5.9

Next, as shown in [Figure 5.10](#), you must specify the vertical sync (refresh) rate. Type the number associated with your choice and press **Enter**. If you're unsure of your monitor's vertical sync range, specify range 1, which is the most conservative. To specify a range other than those listed, you can select choice 5; if you do so, you'll be prompted for the low and high values of the vertical sync range.

Figure 5.10: Specifying the vertical sync rate

 Figure 5.10

You must now specify identification and description strings for your monitor, as shown in [Figure 5.11](#). You can enter any text you like. Press **Enter** after typing each string.

Figure 5.11: Specifying the monitor identification and description strings

 Figure 5.11

Next, you must specify your video card and its characteristics. The explanations provided by `xf86config`, shown in [Figure 5.12](#), point out that you can choose to select your card from a database. However, even if you do so, you'll be given the opportunity to specify non-standard values. Unless you have a specific reason for doing so, you should not override the values in the database. Moreover, you should be careful to choose only the database entry that exactly matches your card; cards having similar model names may have significantly different hardware characteristics.

Figure 5.12: Preparing to examine the card database

 Figure 5.12

[Figure 5.13](#) shows the screen you use to choose your card. Simply type the number associated with your card and press **Enter**. If you suspect that your card appears on a subsequent page, press `q` to page forward through the database. If you accidentally page past your card, simply continue moving forward; when the program reaches the last entries of the database, it cycles back to the beginning.

Figure 5.13: Examining the card database

 Figure 5.13

After you choose your video card, `xf86config` reports your choice. As in [Figure 5.14](#), `xf86config` may provide instructions, such as "Do NOT probe clocks." It's a good idea to write these down so that you remember to observe them even after they've scrolled off the screen.

Figure 5.14: The selected card definition

 Figure 5.14

Next, you must specify the X server you want to use, as shown in [Figure 5.15](#). Consult **Table C.1** to determine the appropriate server. Type the number associated with the server and press **Enter** to continue. If you specify choice 4, you'll be prompted to specify which accelerated server you want to use. If you're in doubt, specify server 3, the `XF86_SVGA` server; unless your video card or monitor are quite old, they're likely to support 256-color SVGA.

Figure 5.15: Specifying the server

 Figure 5.15

Next, as shown in [Figure 5.16](#), `xf86config` asks whether it should change the first line of the `/etc/X11/server` file to point to your server.

Respond by typing *y* and pressing **Enter**.

Figure 5.16: Setting the default server

 Figure 5.16

Now, as shown in [Figure 5.17](#), specify the amount of memory installed on your video card by typing the appropriate number and pressing **Enter**. You can determine the amount of memory by examining the documentation that accompanied your card. If you cannot locate the documentation, try a small value, such as 1024K. Generally, choosing too small a value will merely prevent your card from operating at high resolutions; however, choosing too large a value may cause the card to temporarily malfunction.

Figure 5.17: Specifying the amount of video memory

 Figure 5.17

Just as you previously specified text strings that identify and describe your monitor, you should now specify strings that identify and describe your video card, as shown in [Figure 5.18](#). Press **Enter** after typing each string.

Figure 5.18: Specifying the video card identification and description strings

 Figure 5.18

If you selected an accelerated server, you can now enter the RAMDAC settings, as shown in [Figure 5.19](#). Some SVGA servers also support RAMDACs. If you're not using an accelerated server, you can simply type *q* and press **Enter** to omit specification of a RAMDAC. Otherwise, type the number associated with the RAMDAC used by your card and press **Enter**. Determining the correct number may pose a bit of a puzzle. The descriptions given in

the screen specify RAMDAC chips used on particular cards. If you can conveniently view your card, you can inspect it to see if it contains any of the listed chips. If it's not convenient to view your card, type `q` and press **Enter** to omit specification of a RAMDAC. X will autodetect most RAMDACs, so omitting the specification will not likely impair the performance of your video hardware.

Figure 5.19: Specifying the RAMDAC settings

 Figure 5.19

Next, as shown in [Figure 5.20](#), you can specify the programmable clock chip used by your video card. Most video cards lack such a chip; such cards require a `Clocks` line in the X configuration file. If your video card lacks a programmable clock chip, type `q` and press **Enter** to continue; otherwise type the number associated with your card's programmable clock chip and press **Enter**.

Figure 5.20: Specifying the clock chip

 Figure 5.20

As shown in [Figure 5.21](#), `xf86config` asks you to let it probe your system to determine proper clock timings. If you specified a programmable clock chip, you should omit the probe; type `n` and press **Enter** to continue. You should also omit the probe if you earlier noted that probing is not recommended for your card.

You can sometimes improve the accuracy of the clock timings by running the probe yourself after `xf86config` is done and adding an appropriate `Clocks` line to your X configuration. Consult the X documentation for information on how to do so.

Otherwise, you should let `xf86config` probe your system to determine appropriate clock settings: type `y` and press **Enter** to begin the probe.

WARNING: If `xf86config` probes your system and the screen remains black for more than 30 seconds, immediately cancel the probe by turning off the monitor, pressing **Ctrl-C**, and restoring power to your monitor. If the probe fails, it can *permanently* damage your monitor.

Figure 5.21: Beginning the automatic probe

 Figure 5.21

Next, you can specify the color depths and resolutions in which X will operate, as shown in [Figure 5.22](#). Generally, `xf86config`'s default choices are appropriate: you can type 5 and press **Enter** to continue. However, you can change the resolutions allowed when operating at a given color depth by typing the number associated with the color depth and specifying the desired resolution or resolutions.

Figure 5.22: Specifying the modes

 Figure 5.22

Finally, as shown in [Figure 5.23](#), `xf86config` is ready to write the configuration file it has prepared. Generally, you should let it write the file to `/etc/X11/XF86Config`; simply type `y` and press **Enter**. However, if you prefer, you can type `n` and specify a different directory or filename.

Figure 5.23: Writing the configuration file

 Figure 5.23

Once the file has been written, you're ready to start X.

5.4 Starting and Stopping X

Now that you've configured X by using `xf86config`, you're probably eager to see it work. To start X, type the command:

```
startx
```

Your system's screen should briefly go blank and then you should see X's graphical desktop. **Chapter 6,**

Using the X Window System, will teach you how to use X effectively.

WARNING: If the screen is garbled or remains blank for more than about 30 seconds, your X configuration may be faulty. Immediately turn off your monitor or terminate X by pressing **Ctrl-Alt-Backspace**.

To exit X, click on an unused part of the desktop and a pop-up menu will appear. From the menu, select the Exit, Logout, or Quit menu item. X shuts down, returning you to the familiar text-based interface of the Linux shell prompt.



Learning Debian GNU/Linux

By Bill McCarty

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

6. Using the X Window System

Contents:

[Keyboard Operations](#)

[Mouse Operations](#)

[Window Managers](#)

[Desktops](#)

[Using GNOME](#)

[Using GNOME Applets and Applications](#)

[Configuring GNOME](#)

Using the X Window System means interacting with Linux on several different levels. X itself simply provides the graphics for displaying components of a graphical user interface: X draws the screen, draws objects on the screen, and tracks user input actions such as keyboard input and mouse operations. To organize all of this into familiar objects like windows, menus, and scrollbars, X relies on a separate program called a window manager. A window manager alone won't necessarily assure tight integration between applications running under X; that higher degree of integration comes from something called a desktop environment. While X itself is a single program, X under Linux supports several popular window managers, and two popular desktop environments.

To use X effectively, you'll learn the basic keyboard and mouse operations for communicating with X. If you're like most X users, you'll find it helpful to use a window manager and a desktop with X. You'll learn why window managers and desktops are useful and get help in choosing and setting up a window manager and a desktop.

6.1 Keyboard Operations

Using the keyboard with X closely resembles using the keyboard with Microsoft Windows. X sends your keyboard input to the active window, which is said to have the *input focus*. The active window is usually the window in which you most recently clicked the mouse; however, under some circumstances, it can be the window beneath the mouse cursor.

This chapter refers to your pointing device as a *mouse*. However, like Microsoft Windows, X supports a variety of pointing devices.

Microsoft Windows lets you choose to perform most operations by using the keyboard or mouse. In contrast, X was designed for use with amouse. If your mouse isn't functioning, you'll find it quite challenging or even impossible to use most X programs.

Similarly, X provides a few important functions that you can access only via the keyboard:

- Using virtual consoles
- Switching video modes

In addition, you can use the keyboard to terminate X.

6.1.1 Switching Video Modes

When you configured X, you specified the video modes in which X can operate. Recall that the current video mode determines the resolution and color depth of the image displayed by your monitor - for example 16 bits per pixel color depth and 1024×768 pixels screen resolution.

By pressing **Shift-Alt-+** (using the plus key on the numeric keypad), you command X to switch to the next video mode in sequence. X

treats the video modes as a cycle: If X is operating in the last video mode, this key sequence causes X to return to the first video mode.

The similar key sequence **Shift-Alt--** (using the minus key on the numeric keypad) causes X to switch to the previous video model. If you shift to a video mode that your monitor doesn't support - as demonstrated by a unsteady or garbled image - you can use this key sequence to return to a supported video mode, avoiding the inconvenience of terminating X.

6.1.2 Using Virtual Consoles with X

Even while X is running, you can access the Linux virtual consoles. To switch from graphical mode to a virtual console running in text mode, type **Ctrl-Alt-F *n***, where **F *n*** is a function key and *n* is the number of the desired virtual console. X uses virtual console 7, so only virtual consoles 1-6 are accessible while running X.

To switch from a virtual console back to X, type **Alt-F7**. Nothing is lost when you switch from X to a virtual console or back, so you can move freely between the graphical and text operating modes.

6.1.3 Terminating X

As you learned in the previous chapter, you can terminate X by typing **Ctrl-Alt-Backspace**. X immediately terminates each program running under X, closes each open window, and returns your system to text mode.

This key sequence terminates X abruptly; most window managers support gentler ways of terminating X. You'll learn about these later in this chapter.

While X is running, you cannot use the **Ctrl-Alt-Del** sequence to reboot your system. To reboot your system, you can terminate X and then use the **Ctrl-Alt-Del** sequence, or access a terminal window and enter the command:

```
shutdown -r now
```

The `shutdown` command terminates X and then reboots your system.

6.1.4 Terminal Windows

In Windows, you need not restart in DOS mode simply to have access to the DOS command line. Similarly, in X you need not switch to a virtual console simply to have access to the command line. X enables you to open a terminal window. A terminal window resembles the familiar Microsoft Windows MS-DOS Prompt window; like the Linux shell, it lets you type commands and view command output. Various window managers support different ways of accessing a terminal window.

6.1.5 Pop-Up Menus

The terminal window is just one example of a frequently used program under X that you'll want to access. Most window managers install with a default set of common programs that can be accessed by left- or right-clicking with the mouse on the desktop. Most window managers, for example, let you click on the desktop and select a terminal window program from the pop-up menu that appears. However, the pop-up menu displayed by a window manager may display program names rather than program functions. In this case, you may have some difficulty determining which entry on the pop-up menu corresponds to a terminal program. Many programs that provide terminal windows have names that include the sequences *xt* or *xterm*. Selecting such an entry will probably launch a terminal window. You'll learn more about window managers and how to use them later in this chapter.

6.2 Mouse Operations

Mouse operations under X are similar to mouse operations under Microsoft Windows, although you perform them differently. The most common mouse operations are:

- Copying and pasting text
- Using scrollbars

6.2.1 Copying and Pasting Text

To copy and paste text, you must first mark the text. To do so, you move the mouse cursor to the beginning of the text, press the left mouse button, and drag the mouse across the text to be marked. X automatically copies the marked text into a buffer; you don't need to press **Ctrl-C** or perform any other operation. If you find that you need to change the size of the marked text section, you can press the right mouse button and move the mouse to adjust the marked text.

Some window managers display a pop-up menu when you click the right button, even when the mouse cursor is above text. When using such a window manager, you cannot use the right mouse button to adjust the size of the marked text section.

To paste the text, properly position the insertion point and press the middle mouse button. If your mouse has only two buttons, simultaneously press the left and right buttons to simulate pressing the middle mouse button. You may find that this operation requires a little practice before you get it right, but once you've mastered it you'll find it works almost as well as having a three-button mouse.

6.2.2 Using Scrollbars

Many X programs provide scrollbars that resemble those provided by Microsoft Windows programs. However, the operation of scrollbars

under X differs significantly from that under Microsoft Windows.

To page forward using an X scrollbar, you click the left mouse button on the scrollbar. Clicking near the top of the scrollbar scrolls forward a short distance, as little as a single line. Clicking near the bottom of the scrollbar scrolls the window by a page.

To page backward, you click the right mouse button on the scrollbar. Again, clicking near the top of the scrollbar scrolls a short distance, as little as a single line. Clicking near the bottom of the scrollbar scrolls the window by a page.

Some X programs redefine the operation of scrollbars to correspond to that provided by Microsoft Windows. If a scrollbar doesn't respond as you expect, try using the common Windows manipulations: left click below the scroll box to move forward, left click above the scroll box to move backward, or left drag the scroll box to a desired position.

6.2.3 Virtual Desktop

Under X, your desktop can be larger than the size of your monitor. For example, even if your monitor has a maximum resolution of 800×600, you might have a desktop of 1600×1200 or even 3200×2400. Such a desktop is known as a *virtual desktop*. Some desktop environments, including GNOME, provide a tool called a *pager*, which lets you move around the virtual desktop. The pager provides a thumbnail view of your virtual desktop; by clicking within the thumbnail, you center your actual desktop on the clicked location. Some window managers let you simply move the mouse to the edge of the desktop to scroll the virtual desktop.

6.3 Window Managers

Window managers create the borders, icons, and menus that provide a simple-to-use interface. Window managers also control the look and feel of X, letting you configure X to operate almost any way you desire. Some Linux users who are accustomed to the look and feel of Microsoft Windows 9x use the FVWM window manager to establish a user interface that resembles that of Windows 9x, both in appearance and operation. Other Linux users prefer to avoid anything resembling a Microsoft product. [Table 6.1](#) describes the most popular Linux window managers. For detailed information about a variety of window managers, see the X11.Org web site at <http://www.x11.org/wm/>.

Table 6.1: Popular Window Managers

Window Manager	Description
AfterStep	Resembles the user interface of the NEXT computer (NEXTStep).
BlackBox	A small, simple, efficient window manager. Compatible with KWM.
Enlightenment	A highly configurable window manager.

Window Manager	Description
FVWM	One of the most venerable and popular Linux window managers - small, efficient, and configurable. Can mimic the Microsoft Windows 9x user interface. Not fully compliant with GNOME desktop.
ICEWM	A fast, small window manager especially popular among users of Debian GNU/Linux.
KWM	A window manager that sports an accompanying desktop, KDE. The combination of KWM and KDE provides a robust and efficient user interface. However, KWM includes some non-GPL code, inhibiting its adoption as the de facto standard Linux window manager. Not compliant with GNOME desktop.
SCWM	A window manager that has a powerful configuration language, based on the Scheme dialect of LISP.
WindowMaker	Resembles the user interface of NEXTStep. Compatible with KWM.

At present, the two most important window managers appear to be FVWM and Enlightenment. The next two sections describe these window managers in more detail.

6.3.1 FVWM

FVWM is perhaps the most popular Linux window manager. Several other window managers have borrowed from its code base, so many of its capabilities are found in other window managers. Although FVWM lacks the visual flashiness of more recent window managers, it is robust and highly configurable. However, FVWM is not fully compliant with the GNOME desktop; users who plan to use GNOME may prefer to choose a different window manager.

6.3.2 Enlightenment

Enlightenment is the window manager most often used with the GNOME desktop, which is described in the following section. Although Enlightenment is still under development, many Linux users find it stable enough for everyday use. Apart from being highly configurable, Enlightenment is written using CORBA (Common Object Request Broker Architecture). Programs written in any language can interact with Enlightenment via its CORBA interface.

6.4 Desktops

A desktop is a set of desktop tools and applications. The Microsoft Windows 9x desktop includes applications such as the Windows Explorer, accessories such as Notepad, games such as FreeCell and Minesweeper, and utilities such as the Control Panel and its applets. Although you can run X without a desktop, having a desktop helps you work more efficiently. The two most popular desktops used with X are KDE and GNOME.

6.4.1 KDE

KDE (the K Desktop Environment) is a freely available desktop that includes KWM, the K Window Manager, as an integral component. KDE provides a file manager, a help system, a configuration utility and a variety of accessories and applications, including:

- Games such as Kmines, Kpoker, and Ktetrtris
- Graphical applications such as Kfract, a fractal generator, and Kview, an image viewer
- Multimedia applications such as Kmix, a sound mixer, and Kmedia, a media player
- Network applications such as Kmail, a mail client, Knu, a network utility, and Krn, a news client

New KDE accessories and applications are available almost weekly. Work is underway on a complete open source office suite (KOffice) that runs under KDE. You can learn more about KDE and the status of KOffice by browsing the KDE web site at <http://www.kde.org/>.

Although KDE is freely redistributable, KDE uses the QT widget set to create user interface controls. This presents a problem, because QT is distributed under a non-free license that many developers dislike.

Therefore, the most popular desktop environment in the U.S. has been, and continues to be, GNOME rather than KDE.

6.4.2 GNOME

GNOME is a freely available desktop that can be used with any of several window managers, including Enlightenment. Unlike KDE, GNOME is open source software.

One of GNOME's most interesting features is session awareness. When you re-enter GNOME, it reconfigures your desktop to match the state at the time you exited, by launching each application that was open when you exited. GNOME even restores each application to its former state by, for example, moving to the page that was open when you exited.

GNOME provides desktop tools similar to those of KDE, including:

- Games such as FreeCell, Gnobots, Gnometriz, and Gnome Mines
- The GNU Image Manipulation Program (GIMP)
- Network applications such as Mailman, which helps you track your mailing lists; Talk, which lets you exchanged typed messages with another user in real time; and Synchronize, which lets you synchronize files on multiple systems
- Multimedia applications such as Audio Mixer and CD Player
- General applications such as gEdit, a text editor, Netscape Navigator, a Linux version of the popular browser, and Gnumeric, a spreadsheet
- Utilities for configuring GNOME and your Linux system

GNOME developers, like KDE developers, release new applications regularly. Check the GNOME Web site at <http://www.gnome.org/> for the latest information.

6.5 Using GNOME

In this section, you'll learn how to configure and use the GNOME desktop and the Enlightenment window manager. If you choose to use a different desktop or window manager, you should consult the documentation that accompanies each. However, you should read this section anyway, because the procedures for configuring various desktops and window managers are more similar than different: the way you perform each step may vary but the function of each step will not.

6.5.1 Launching GNOME and Enlightenment

Before starting GNOME, you must configure the X startup files. Login as `root`, move to the `/etc/X11` directory, and enter the following command:

```
cp Xsession Xsession.SAVE
```

This command makes a copy of your *Xsession* file so that you can restore it to its current state if something goes wrong. If you get an error informing you that the *Xsession* file doesn't exist, simply ignore the error. Next, using `ae` or another text editor of your choice, edit your *Xsession* file to contain these lines at the beginning of the file:

```
#!/bin/bash
xterm &
gmc &

window-manager &
panel
exit 0
##
```

[Table 6.2](#) gives possible values for *window-manager*, which lets you specify which window manager you want to use. You must have installed the proper package containing the window manager you select.

Table 6.2: Window Manager Program Path Names

Window Manager	Path Name
Enlightenment	<i>/usr/bin/X11/enlightenment</i>
FVWM	<i>/usr/bin/X11/fvwm95</i>
FVWM95	<i>/usr/bin/X11/fvwm2</i>
ICEWM	<i>/usr/bin/X11/icewm-gnome</i>
TWM	<i>/usr/bin/X11/twm</i>
Window Maker	<i>/usr/bin/X11/WindowMaker-gnome</i>

To start GNOME, type the command:

```
startx
```

You should see the GNOME desktop, as shown in [Figure 6.1](#). The contents of your own desktop may be different, of course.

Figure 6.1: The GNOME desktop

Figure 6.1

6.5.2 Logging Out

To log out of GNOME, left click on the main menu, which resembles a foot, as shown in [Figure 6.2](#). From the pop-up menu that appears, select the Log Out menu item. A Log Out dialog box, shown in [Figure 6.3](#), appears and asks you to confirm your decision to log out. Selecting Yes terminates your GNOME session.

Figure 6.2: Logging out of GNOME

Figure 6.2

Figure 6.3: The log out dialog box

 Figure 6.3

6.5.3 Parts of the Display

[Figure 6.4](#) shows the parts of the GNOME display, which are described in the following sections.

Figure 6.4: Parts of the GNOME desktop

 Figure 6.4

6.5.3.1 Home directory icon

The home directory icon normally appears in the upper left corner of the display and resembles a file folder in appearance. The icon provides a convenient way to access the file manager: double clicking the icon with the left mouse button launches the file manager, which displays the contents of the user's home directory.

6.5.3.2 Desktop

The desktop is the empty area of the display, where no windows or icons appear. Clicking the desktop with the middle mouse button causes a pop-up menu to appear; the menu lets you conveniently launch popular applets and applications. Right clicking the desktop causes a different pop-up menu to appear; this menu lets you arrange the desktop windows and icons.

6.5.3.3 Drive icon

If you have permission to mount a drive, your desktop will include an icon representing the drive. If you right click on the icon, a pop-up menu appears. The menu lets you mount the device, eject the device's media, or open a file manager window to view the device.

6.5.3.4 Panel

The panel appears along the bottom edge of the display. However, if you prefer a different location, you can move the panel. The panel resembles the Windows 9x taskbar: You can use it to launch programs, switch from one program to another, and perform other tasks.

The panel normally contains the main menu, the pager, and two hide buttons. However, your panel may not initially display the pager.

The panel can also contain applets, programs represented as panel icons. Applets are typically small programs that display information or take action when clicked. For example, a launcher applet launches an application when clicked.

6.5.3.5 Date & time applet

The date and time applet displays your system's current date and time. If the date and time applet is not visible, you can add it to the panel in much the same way you add the pager to the panel. Simply select Panel → Add Applet → Utility → Clock from the main menu. Once you've added the clock applet to the panel, it will appear automatically the next time you start GNOME.

6.5.3.6 Main menu

The main menu resembles a big foot. Left clicking the main menu presents a menu from which you can choose a variety of programs. Several of the menu items are submenus; selecting such a menu item pops up a new menu to the side of the original menu item.

6.5.3.7 Pager

The pager lets you switch between running programs and navigate the desktop. If you don't see the pager on your panel, you can launch the

pager by using the main menu, as shown in [Figure 6.5](#): simply select Panel → Add Applet → Utility → Gnome Pager. Once you've launched the pager, it will automatically appear the next time you start GNOME.

Figure 6.5: Launching the pager

 Figure 6.5

The pager has two parts: The left part, which consists of a matrix of rectangles lets you navigate the desktop; the right part, which consists of a matrix of icons and text representing running tasks, lets you switch tasks. To switch to a task, left click the pager icon corresponding to the task.

Depending on your X configuration, GNOME may provide a virtual desktop larger than your monitor can display. In this case, the pager lets you switch between pages of the desktop. For example, [Figure 6.4](#) shows that GNOME has provided a virtual desktop that has four pages, only one of which is visible at a time. The highlighted icon shows the page you're currently viewing as your desktop. To view a different page, simply left click the icon that represents the desktop page you want to view.

6.5.3.8 Hide button

You can hide and restore the panel by left clicking a hide button. Hiding the display is useful when you're using a window manager that likes to display a task bar or other information along the bottom of the screen.

6.5.3.9 Help viewer

The GNOME help viewer works much like a familiar web browser, except that you can use it to view primarily help information, not web pages. GNOME launches the help viewer whenever you select the

Help menu item of an application or applet. You can also launch the help viewer by selecting the Help System menu item from the main menu.

The default home page of the help viewer includes a hyperlink that takes you to the GNOME User's Guide. The user's guide will help you discover additional useful GNOME features and capabilities.

6.5.3.10 File Manager

The file manager resembles the Windows 9x Explorer in both appearance and function. To launch the file manager, double click a directory or drive icon, or select File Manager from the main menu. The next section briefly explains the operation of the file manager.

6.5.4 Using the File Manager

Like the Microsoft Windows Explorer, the file manager window has two main panes. As shown in [Figure 6.6](#), the left file manager pane presents a hierarchical directory tree whereas the right pane shows the contents of the directory currently selected in the left pane. To select a directory, simply left click it.

Figure 6.6: The file manager

 Figure 6.6

The right pane can show an icon for each file or detailed information about each file, as shown in [Figure 6.7](#). To switch from icon to detailed mode, left click the proper button on the file manager toolbar. You can also select custom mode, which lets you tailor the display appearance according to your own taste.

Figure 6.7: The file manager in detailed mode

 Figure 6.7

Like the Windows Explorer, the file manager can copy, move, rename, and delete files. To perform one of these operations, you must first select the file or files you want to copy, move, rename, or delete. You can select a single file by left clicking in the right file manager pane. To select additional files, hold down the **Ctrl** key as you select them. Alternatively, you can click and drag the cursor around a group of files. The Edit menu provides additional ways of selecting files.

To move a file, simply drag it to its new location. To copy a file, hold down **Shift** while dragging it. Alternatively, you can right click on a file and use the pop-up menu to specify the action you want to perform. The file manager then displays a dialog box that lets you specify additional options.

To rename a file, right click on the file and select Properties from the pop-up menu. Simply type the new name in the File Name field and click on OK.

To delete a file, right click on the file and select Delete from the pop-up menu. A dialog box asks you to confirm your decision.

WARNING: Bear in mind that Linux provides no recycle bin for deleted files; once you delete a file you'll be unable to recover it except from a backup copy.

The file manager lets you double click on a file to launch the application associated with the file. Alternatively, you can right click on the file and select Open With from the pop-up menu. GNOME launches a dialog box that lets you specify the application that should be launched.

Many applications are GNOME compliant, supporting drag and drop operations like those supported by Windows 9x. For example, you can open two file manager windows and drag and drop files or directories between them.

The file manager menus provide additional functions, including the ability to configure the operation of the file manager. If you're familiar with the Windows Explorer, you'll find most of these functions and capabilities familiar. Consult the GNOME User's Guide for further information about the GNOME file manager.

6.6 Using GNOME Applets and Applications

The default GNOME installation includes several applets and applications. This section briefly explains two of the most popular GNOME programs, the GNOME terminal application and the GNOME CD Player applet.

6.6.1 GNOME Terminal

The GNOME terminal application, shown in [Figure 6.8](#), provides a window in which you can type shell commands and view their output. To launch GNOME terminal, select Utilities → GNOME Terminal from the main menu. You can open multiple GNOME terminal windows if you like.

Figure 6.8: Editing terminal settings

Figure 6.8

The Settings menu lets you configure the operation of GNOME terminal. For example, you may find that the default font is too large or too small for your liking. If so, select Settings → Preferences from the GNOME terminal window. A dialog box pops up. Select the General tab and left click the Browse button next to the Font field. A second dialog box pops up, from which you can select the font, font style, and font size you prefer.

To exit GNOME terminal, simply type `exit` on the command line and press **Enter**. Alternatively, select File → Close Terminal from the menu.

6.6.2 GNOME CD Player

[Figure 6.9](#) shows the GNOME CD Player, which is represented by a window on the desktop and an icon on the panel. CD Player lets you

play audio CDs though your computer's sound card. However, CD Player won't function unless you have read access to your CD-ROM's device file. If CD Player fails, log in as root and use the following command to give all users read access to the CD-ROM:

```
chmod a+r /dev/cdrom
```

This command assumes that the symbolic link */dev/cdrom* correctly refers to your CD-ROM device; if necessary, use a different argument that refers to your CD-ROM device.

Figure 6.9: The GNOME CD Player

 Figure 6.9

The buttons that operate CD Player resemble those found on other CD Players, such as the one provided with Windows 9x. You can play, stop, or pause the CD Player, eject the CD, fast forward or rewind.

If your computer is connected to the Internet, CD Player can also access the database of CD information help on the CDDb Web site, <http://www.cddb.org/>. By doing so, CD Player can determine the artist and title of a CD and the titles of its tracks.

CD Player also lets you open a dialog box that lets you manually edit CD information. This is helpful if your computer is not connected to the Internet or if you find that the information on the CDDb database is incomplete or not to your liking.

6.7 Configuring GNOME

Like most GNOME applications, GNOME itself is highly configurable. You can configure GNOME's panel, its main menu, and its overall appearance and function. The following sections briefly show you how.

6.7.1 The GNOME Panel

You can add a launcher to the GNOME panel. Clicking on a launcher launches a predetermined application. To add a new launcher applet, right click on the panel and select Add New Launcher from the pop-up menu. The Create Launcher Applet dialog box appears, as shown in [Figure 6.10](#).

Figure 6.10: Creating a launcher applet

 Figure 6.10

You can specify a name for the launcher, a comment, and the command that GNOME executes to launch the application. GNOME will automatically provide a default icon, but you can specify the icon of your choice by clicking the Icon button.

If an application is already on the main menu, you can quickly create a launcher for it. Simply right click on the application's menu item and select Add This Launcher To Panel from the pop-up menu.

If your panel contains many launchers, it may become crowded and confusing. To remedy this, you can create one or more drawers, like that shown in [Figure 6.11](#). Drawers act like menus; you click on a drawer to open it and view the launchers it contains. Clicking an open drawer closes it and removes its contents from sight.

Figure 6.11: A drawer

 Figure 6.11

To add a drawer, right click on the panel and select Add Drawer from the pop-up menu. To move a launcher into the drawer, right click on the launcher and select Move Applet from the pop-up menu. Move the cursor over the drawer and click the left mouse button.

If you add a launcher or drawer and later decide you don't want it, you can remove it from the panel. Simply right click on the unwanted applet and select Remove From Panel from the pop-up menu, as shown in [Figure 6.12](#).

Figure 6.12: Removing a panel item

 Figure 6.12

6.7.2 The GNOME Main Menu

You can configure GNOME's main menu by using the menu editor. To launch the menu editor, select Settings → Menu Editor from the main menu, as shown in [Figure 6.13](#).

Figure 6.13: Launching the menu editor

 Figure 6.13

The menu editor window, shown in [Figure 6.14](#), has two main panes. Its appearance and operation resemble that of the file manager. The left pane of the menu editor hierarchically displays the menu tree, whereas the right pane shows information pertaining to the currently selected menu item. You can use toolbar buttons to move the current menu item up or down the menu tree, add a new submenu or menu item, or delete the current menu item.

Figure 6.14: Using the menu editor

 Figure 6.14

6.7.3 The GNOME Control Center

You can configure the appearance and operation of GNOME and GNOME-compliant applications by using the GNOME control center, shown in [Figure 6.15](#). The function of Control Center resembles that of the Windows 9x control panel, though it looks different and works somewhat differently. To launch the control center, select Settings → GNOME Control Center from the GNOME main menu.

Figure 6.15: The GNOME Control Center

 Figure 6.15

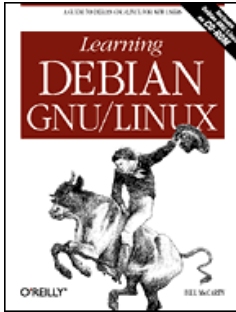
Like the Windows control panel, which uses small programs called applets to perform its functions, the GNOME control center uses small programs called capplets. However, the control center's user interface hides this detail from you, so you needn't normally be aware of what's happening behind the scenes. The control center user interface resembles that of file manager and menu editor: The left pane of the control center window presents a hierarchically structured set of configuration categories and the right pane displays information pertaining to the current choice.

Using control center, you can:

- Select background properties
- Configure a screen saver
- Select a desktop theme
- Select a window manager
- Configure the default text editor

- Specify MIME types that control the handling of multimedia files
- Configure the keyboard bell and sounds
- Configure keyboard and mouse properties
- Specify applications that GNOME automatically launches when it starts
- Specify a variety of options governing the appearance of GNOME-compliant applications

Simply select the configuration category by clicking in the left pane. You can then revise the configuration parameters by specifying the desired values in the right pane. The buttons that appear in the right pane vary from caplet to caplet. The Try button lets you experiment without permanently altering the GNOME configuration. The OK button permanently updates the GNOME configuration whereas the Cancel button discards your changes.



Learning Debian GNU/Linux

By Bill McCarty

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

7. Configuring and Administering Linux

Contents:

[User and Group Administration](#)

[Filesystem Administration](#)

[Starting and Stopping the System and Services](#)

[Viewing System Messages and Logs](#)

[Setting the System Time and Time Zone](#)

This chapter equips you to perform common system administration tasks. By using a few simple commands and a text editor, you can override many configuration choices made during system installation. You'll learn how to add, delete, and modify user accounts, and how to add and delete groups and change their membership. You'll learn how to configure swap space and how to cause Linux to automatically mount filesystems. This chapter doesn't deal with the configuration and administration of network facilities, such as servers. Those topics are covered in Chapters 10, 11, and 12.

7.1 User and Group Administration

In this section, you'll learn how to perform common administrative tasks affecting users and groups. Most system administration tasks require that you login as `root`. Throughout this section and subsequent section, you should assume that you must be logged in as `root`, unless directed otherwise.

7.1.1 Creating a User Account

To create a user account, you use the `adduser` command, which has the form:

```
adduser  
userid
```

where `userid` specifies the name of the user account that you want to create. The command prompts you for the information needed to create the account.

Here's a typical example of using the command, which creates a user account named `newbie`:

```
debian:~#  
  
adduser newbie  
Adding user newbie...  
Adding new group newbie (1001).  
Adding new user newbie (1001) with group newbie.  
Creating home directory /home/newbie.  
Copying files from /etc/skel  
Changing password for newbie  
Enter the new password (minimum of 5, maximum of 8 characters)  
Please use a combination of upper and lower case letters and numbers.  
Re-enter new password:  
Password changed.  
Changing the user information for newbie  
Enter the new value, or press return for the default  
  
Full Name []:  
  
Newbie Dewbie  
Room Number []:  
Work Phone []:  
Home Phone []:  
Other []:  
Is the information correct? [y/n]  
  
y  
debian:~#
```

Notice that the lines where the password was typed were overwritten by the subsequent lines. Moreover, for security, passwords are not echoed to the console as they are typed.

Notice also that several of the information fields were omitted - for example, Room Number. You can specify such information if you think it may be useful, but the system makes no use of the information and doesn't require you to provide it.

The similarly named `useradd` command also creates a user account, but does not prompt you for the password or other information.

When the command establishes a user account, it creates a home directory for the user. In the previous example, the command would have created the directory `/home/newbie`. It also places several configuration files in the home directory, copying them from the directory `/etc/skel`. These files generally have names beginning with the dot (.) character, so they are hidden from an ordinary `ls` command. Use the `-a` argument of `ls` to list the names of the files. The files are generally ordinary text files, which you can view with a text editor, such as `ae`. By modifying the contents of such files, you can control the operation of the associated application. For example, the `.bashrc` file controls the operation of the BASH shell, which you'll learn more about in **Chapter 13**,

Conquering the BASH Shell.

7.1.2 Changing a User's Name

You can change the name associated with a user account, by using the `chfn` command:

```
chfn -f
name userid
```

where *name* specifies the new name and *userid* specifies the account to be modified. If the name contains spaces or other special characters, it should be enclosed in double quotes ("). For example, to change the name associated with the account `newbie` to Dewbie Newbie, you would enter the following command:

```
chfn -f "Dewbie Newbie" newbie
```

7.1.3 Changing a User Account Password

From time to time, you should change your password, making it more difficult for others to break into your system. As system administrator, you may sometimes need to change the password associated with a user's account. For instance, some users have a bad habit of forgetting their password. They'll come to you, the system administrator, seeking help in accessing their account.

To change a password, you use the `passwd` command. To change your own password, enter a command like this one:

```
passwd
```

This command changes the password associated with the current user account. You don't have to be logged in as `root` to change a password. Because of this, users can change their own passwords without the help of the system administrator. The `root` user, however, can change the password associated with any user account, as you'll see shortly. Of course, *only* `root` can do so - other users can change only their own password.

The `passwd` command initiates a simple dialog that resembles the following:

```
$
```

passwd

```
Changing password for newbie
```

```
Old password:
```

```
Enter the new password (minimum of 5, maximum of 8 characters)
```

```
Please use a combination of upper and lower case letters and numbers.
```

```
New password:
```

```
Re-enter new password:
```

```
Password changed.
```

Notice the restrictions governing the choice of password, which are designed to prohibit passwords that might be easily guessed. If you choose a password that violates these restrictions, the command will refuse the password, prompting you for another.

As the `root` user, you can change the password associated with any user account. The system doesn't ask you for the current password, it immediately prompts for the new password:

```
debian:~# passwd newbie
Changing password for newbie
Enter the new password (minimum of 5, maximum of 8 characters)
Please use a combination of upper and lower case letters and numbers.
New password:
Re-enter new password:
Password changed.
```

Information on users is stored in the file */etc/passwd*, which you can view using a text editor. Any user can read this file, though only the `root` user can modify it. If you selected shadow passwords, passwords are encrypted and stored in the file */etc/shadow*, which can be read only by the `root` user.

7.1.4 Configuring Group Definitions

Recall from **Chapter 4**,

Issuing Linux Commands that Linux uses groups to define a set of related user accounts that can share access to a file or directory. You probably won't often find it necessary to configure group definitions, particularly if you use your system as a desktop system rather than a server. However, when you wish, you create and delete groups and modify their membership lists.

7.1.4.1 Creating a group

To create a new group, use the `groupadd` command:

```
groupadd  
group
```

where *group* specifies the name of the group to be added. Groups are stored in the file */etc/group*, which can be read by any user but modified only by `root`.

For example, to add a group named `newbies`, you would enter the following command:

```
groupadd newbies
```

7.1.4.2 Deleting a group

To delete a group, use the `groupdel` command:

```
groupdel  
group
```

where *group* specifies the name of the group to be deleted. For example, to delete the group named `newbies`, you would enter the following command:

```
groupdel newbies
```

7.1.4.3 Adding a member to a group

To add a member to a group, you use a special form of the `adduser` command:

```
adduser  
user group
```

where *user* specifies the member and *group* specifies the group to which the member is added. For example, to add the user `newbie01` to the group `newbies`, you would enter the following command:

```
adduser newbie01 newbies
```

7.1.4.4 Removing a member from a group

Unfortunately, no command removes a user from a specified group. The easiest way to remove a member from a group is by editing the */etc/group* file. Here's an excerpt from a typical */etc/group* file:

```
users:x:100:  
nogroup:x:65534:  
bmccarty:x:1000:  
newbies:x:1002:newbie01,newbie02,newbie03
```

Each line in the file describes a single group and has the same form as other lines, consisting of a series of fields separated by colons (:). The fields are:

Group name

The name of the group.

Password

The encrypted password associated with the group. This field is not generally used, containing an *x* instead.

Group ID

The unique numeric ID associated with the group.

Member list

A list of user accounts, with a comma (,) separating each user account from the next.

To remove a member from a group, first create a backup copy of the */etc/group* file:

```
cp /etc/group /etc/group.SAVE
```

The backup can prove helpful if you modify the file incorrectly. Next, open the */etc/group* file in a text editor. Locate the line that describes the group and delete the user name and the following comma, if any. Save the file, exit the editor, and check your work.

7.1.5 Deleting a User Account

To delete a user account, use the `userdel` command:

```
userdel  
user
```

where *user* specifies the account to be deleted. If you want to delete the user's home directory, its files and subdirectories, use this form of the command:

```
userdel -r  
user
```

WARNING: Because deleted files can't generally be recovered, you should backup potentially useful files before deleting a user account.

7.1.6 Configuring Access to Shells

The BASH shell, which you met in **Chapter 4**, is the most popular, but not the only Linux shell. Others include:

ash

a version of the `sh` shell that resembles the System V shell

cs

the C shell, favored by many users for interactive use

ksh

the Korn shell, the third major Unix shell

sh

the Bourne shell, a precursor of BASH, also known as the `bs` shell

tcsh

an enhanced version of `cs`

zsh

the Z shell, a feature-packed version of the Korn shell

When you create a new user, the system automatically assigns the shell (command interpreter) that Linux presents to the user when the user logs in. Debian GNU/Linux assigns the BASH shell, as specified by the file */etc/adduser.conf*. However, you can assign another shell, if you prefer. The shell must be on the list of available shells, which resides in the file */etc/shells*.

7.2 Filesystem Administration

When Linux starts, it automatically mounts the file systems specified in the file */etc/fstab*. By revising this file, you can customize the operation of your system.

7.2.1 Configuring Local Drives

When you install Linux, the installation program configures the file */etc/fstab* to specify what filesystems are to be mounted when the system is started. Here's a typical */etc/fstab* file:

```
# /etc/fstab: static file system information.
#
# <file system> <mount point> <type> <options>          <dump> <pass>
/dev/hda2      /                ext2    defaults          0        1
/dev/hda3      none            swap    sw                0        0
proc           /proc           proc    defaults          0        0
```

The first three lines, those beginning with a hash mark (#), are comments that are ignored by the system; they merely help human readers identify and understand the file. The next three lines each specify a filesystem to be mounted at system startup. Six columns of information appear:

Filesystem

The device that contains the filesystem.

Mount point

The system directory that will hold the filesystem.

Filesystem type

Specifies the type of the filesystem. Popular types include:

ext2

the standard Linux filesystem

swap

the standard Linux swap filesystem

proc

a special filesystem provided by the kernel, used by system components to obtain system information in a standard way

iso9660

the standard filesystem used on CD-ROM

msdos

the standard MS-DOS filesystem

See the man page for `mount` for other filesystem types.

Mount options

Specifies the options given when the filesystem is mounted. If multiple options are given, each is separated from the next by a comma (,); no spaces appear within the list of options. Popular options include:

defaults

Specifies a series of options appropriate for most filesystems. For details, see the man page for `mount`.

errors=remount-ro

Specifies that if errors are found when the filesystem is checked, the filesystem will be remounted in read-only mode so that the system administrator can analyze the errors without risking further damage.

sw

Specifies that the filesystem will be mounted as a swap partition.

ro

Specifies that the filesystem will be mounted for read access only. This option is always specified for CD-ROM devices and may be specified for other devices.

noauto

Specifies that the filesystem will not be automatically mounted at system startup.

In addition, the *user* option can be specified. This option allows any user - not only `root` - to mount the filesystem.

Dump flag

Specifies whether the `dump` command will create a backup of the filesystem. Filesystems with no value or a value of zero will not be dumped.

Pass

Specifies the order in which filesystems are checked at boot time. No value or a value of zero specifies that the filesystem will not be checked.

You can modify the lines within the `/etc/fstab` file and add new lines as you see fit. For example, here's a line that specifies a CD-ROM drive:

```
/dev/cdrom      /cdrom          iso9660 ro
```

By adding this line to the `/etc/fstab` file, you instruct the system to mount the CD-ROM filesystem when the system starts. If you don't want the filesystem automatically mounted, you can specify this line:

```
/dev/cdrom      /cdrom          iso9660 ro,noauto
```

The system will not automatically mount the CD-ROM filesystem described by this line, but you can mount the CD-ROM by using the `mount` command. Because the system already knows the device, mount point, filesystem type, and options, you can abbreviate the mount command to:

```
mount /cdrom
```

or:

```
mount /dev/cdrom
```

Either of these is equivalent to:

```
mount -t iso9660 -o ro /dev/cdrom /cdrom
```

You can automatically mount additional hard disk partitions by describing them in the `/etc/fstab` file:

```
/dev/hdb1      /home           ext2    defaults
```

Another tip is to use an entry in the `/etc/fstab` file to allow users other than `root` to mount a floppy disk:

```
/dev/fd0       /floppy         auto    noauto,user
```

7.2.2 Configuring Swap Partitions

Just as you can use the `mount` and `unmount` commands to explicitly mount and unmount filesystems, you can control the operation of swap partitions by using the `swapon` and `swapoff` commands.

If you want to modify your swap partition, you may need to temporarily turn off swapping. To do so, enter the command:

```
swapoff -a
```

This command turns off swapping on every swap device mentioned in */etc/fstab*. If you want to turn off swapping on a particular device, enter the command:

```
swapoff /dev/  
device
```

where *device* specifies the swap device; for example, *hda3*.

To turn on swapping, enter the command:

```
swapon -a
```

This command turns on swapping for all swap devices mentioned in */etc/fstab*. If you want to turn on swapping on a particular device, enter the command:

```
swapon /dev/  
device
```

where *device* specifies the swap device; for example, *hda3*.

7.3 Starting and Stopping the System and Services

Using Linux commands, you can start and stop the system or start and stop services, such as the Apache web server.

7.3.1 Starting and Stopping the System

Generally, you start your system by turning on its power. However, you can use a Linux command to cause a system to restart itself. If you enter the command:

```
shutdown -r now
```

the system will immediately begin to shut down. Once it's shut down, it will reboot. If you want to provide a delay before commencing the shutdown, use this form of the command:

```
shutdown -r +  
mm
```

where *mm* gives the number of minutes until the shutdown commences. The command displays a message to system users who have active shells displaying a command prompt. After commencement of a shutdown, users cannot initiate new login sessions.

If you want to halt the system, that is, shut down the system without causing it to reboot, use the command:

```
shutdown -h now
```

7.3.2 Starting and Stopping Services

Services are daemon programs that run without an associated console. They listen for network connections from clients, which request them to perform an action or provide information. [Table 7.1](#) describes some of the most important services.

Table 7.1: Important Services

Service	Function
apache	Web server
atd	Runs commands at predefined times.

Service	Function
cron	Runs commands at predefined times; offers more flexibility than <code>atd</code> .
exim	Mail transfer agent.
gpm	Provides cut and paste to virtual consoles.
lpd	Controls the printer.
netbase	Basic networking services (<code>inetd</code> and <code>portmap</code>).
netstd_init	Network routing (<code>routed</code>).
netstd_misc	Miscellaneous networking services.
nfs-server	Network file system (<code>nfsd</code>).
samba	Microsoft-compatible networking (<code>smbd</code> and <code>nmdbd</code>).

If a network services fails, you may want to restart it without rebooting your system. To do so, you can enter a command such as this:

```
/etc/init.d/  
service start
```

where `service` gives the name of the service, as shown in [Table 7.1](#).

If you want to stop a service, use a command such as this:

```
/etc/init.d/  
service stop
```

If a service is behaving erratically, you may be able to stabilize it by stopping and restarting it:

```
/etc/init.d/  
service stop  
/etc/init.d/  
service start
```

Pause a few seconds before entering the start command to give the service time to come to a smooth stop.

7.4 Viewing System Messages and Logs

Linux maintains several system logs that help you administer a Linux system by informing you of important events. Probably the most important log is the file */var/log/messages*, which records a variety of events, including system error messages, system startups, and system shutdowns. Like most other Linux files, the file contains ASCII text, so you can view it with a text editor or the text processing commands described in **Chapter 13**.

A special command, `dmesg`, makes it easy to view the log messages related to the most recent system startup. If your system is behaving unusually, use `dmesg` to quickly see if something went wrong during the system startup sequence. Of course, you must have some way of determining what's usual and unusual among the many messages emitted during system startup. The best way to do so is to print the output of the `dmesg` command and keep it on hand for comparison with suspicious output. If your system has an attached printer, you can print the output of `dmesg` by entering the following command:

```
dmesg | lpr
```

Other logs found in the */var/log* directory include:

apache

A directory that contains two log files pertaining to the Apache web server, *access.log* and *error.log*.

exim

A directory that contains several log files pertaining to the `exim` mail transfer agent.

nmb and smb

Files that contain log entries pertaining to Samba, the Microsoft-compatible networking server.

ppp.log

A file the contains log entries pertaining to PPP.

7.5 Setting the System Time and Time Zone

Linux provides commands that let you set the current system date and time and the system time zone.

7.5.1 Setting the Current System Date and Time

To display the current system time, enter the `date` command:

```
debian:~# date
Fri Jul 30 02:28:22 PDT 1999
```

To set the current system time, use the following form of the `date` command:

```
date
MMDDhhmm[
CC]
YY[.
ss]
```

The parts of the command argument have the following meanings:

MM

A two-digit month, 01-12.

DD

A two-digit day of month, 01-31.

hh

A two-digit hour, 00-24.

mm

A two-digit minute, 00-59.

CC

An optional two-digit century; for example, 19 or 20.

YY

A two-digit year; for example, 99 or 00.

ss

An optional two-digit second, 00-59.

The command displays the time you enter and then sets the system time:

```
debian:~# date 073010411999
Fri Jul 30 10:41:00 PDT 1999
```

When you power down or reboot your system, the system date and time will revert to the values held in non-volatile (CMOS) memory. To store the Linux date and time in CMOS, issue the following command:

```
hwclock --systohc
```

If you set your clock to UTC rather than local time, issue the command:

```
hwclock --systohc --utc
```

7.5.2 Setting the Time Zone

To set the time zone, use the `tzconfig` command. The command initiates a dialog that displays the current time zone and asks if you want to change it. If you reply *yes*, the command prompts you to choose from a list of geographical areas and then cities. Based on your choices, the command sets and displays the current time zone. A typical dialog resembles the following:

```
debian:~#
```

tzconfig

```
Your current time zone is set to US/Pacific
Do you want to change that? [n]:
```

y

```
Please enter the number of the geographic area in which you live:
```

- | | |
|----------------------|-----------------------------------|
| 1) Africa | 7) Australia |
| 2) America | 8) Europe |
| 3) US time zones | 9) Indian Ocean |
| 4) Canada time zones | 10) Pacific Ocean |
| 5) Asia | 11) Use System V style time zones |
| 6) Atlantic Ocean | 12) None of the above |

Then you will be shown a list of cities which represent the time zone in which they are located. You should choose a city in your time zone.

```
Number:
```

2

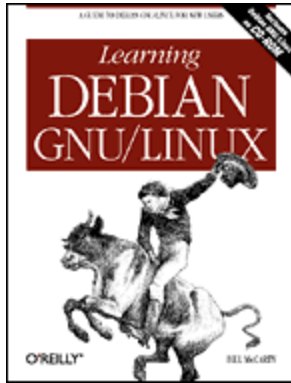
```
Adak Anchorage Anguilla Antigua Araguaina Aruba Asuncion Atka Barbados
Belem Belize Bogota Boise Buenos_Aires Cancun Caracas Catamarca Cayenne
Cayman Chicago Chihuahua Cordoba Costa_Rica Cuiaba Curacao Dawson
Dawson_Creek Denver Detroit Dominica Edmonton El_Salvador Ensenada
Fort_Wayne Fortaleza Glace_Bay Godthab Goose_Bay Grand_Turk Grenada
Guadeloupe Guatemala Guayaquil Guyana Halifax Havana Indiana/Indianapolis
Indiana/Knox Indiana/Marengo Indiana/Vevay Indianapolis Inuvik Iqaluit
```

Jamaica Jujuy Juneau Knox_IN La_Paz Lima Los_Angeles Louisville Maceio
Managua Manaus Martinique Mazatlan Mendoza Menominee Mexico_City Miquelon
Montevideo Montreal Montserrat Nassau New_York Nipigon Nome Noronha Panama
Pangnirtung Paramaribo Phoenix Port-au-Prince Port_of_Spain Porto_Acre
Porto_Velho Puerto_Rico Rainy_River Rankin_Inlet Regina Rosario Santiago
Santo_Domingo Sao_Paulo Scoresbysund Shiprock St_Johns St_Kitts St_Lucia
St_Thomas St_Vincent Swift_Current Tegucigalpa Thule Thunder_Bay Tijuana
Tortola Vancouver Virgin Whitehorse Winnipeg Yakutat Yellowknife

Please enter the name of one of these cities or zones
You just need to type enough letters to resolve ambiguities
Press Enter to view all of them again
Name []:

Los_Angeles

Your default time zone is set to 'America/Los_Angeles'.
Local time is now: Fri Jul 30 02:26:32 PDT 1999.
Universal Time is now: Fri Jul 30 09:26:32 UTC 1999.



Learning Debian GNU/Linux

By Bill McCarty

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

8. Using Linux Applications and Clients

Contents:

[Linux Desktop Applications](#)

[Other Approaches to Desktop Computing](#)

This chapter describes how you can use Linux for common desktop computing tasks. You'll learn about popular desktop suites and applications. This chapter also introduces VMWare, a program that lets you run Microsoft Windows and Linux simultaneously, and WINE, a program that lets you run some Microsoft applications under Linux.

The Debian Project is firmly committed to including only truly Open Source software in its main distribution. Not all the programs mentioned in this chapter and the next meet the Open Source definition. Part of the beauty of Linux, though, is the ease with which it can be enhanced. The Contributed and Non-free archives on the Debian site include a wealth of additional software, and some programs are available direct from third party providers.

8.1 Linux Desktop Applications

Not long ago, running Linux meant abandoning your easy-to-use WYSIWYG word processor or configuring your system for dual boot, so that you could boot Microsoft Windows or IBM OS/2 to access friendly and familiar desktop applications. At that time, Linux supported no robust, full-featured graphical desktop suites.

Today, Linux supports several desktop suites; several more are under development. Apparently, if rumors are to be believed, even Microsoft is considering adapting their Office desktop suite to run under Linux. Whether or not Linux users can expect to soon run Microsoft Office for Linux (or whatever name Microsoft might give this potential product), Linux users who prefer to avoid other operating systems can now do so without compromising their ability to produce first-rate documents, spreadsheets, and graphics.

This section describes the three most popular desktop applications for Linux:

- Applix Applixware
- Corel WordPerfect for Linux
- Sun's StarOffice

In addition, this section briefly describes other desktop options:

- Angoss SmartWare
- Axene Xclamation
- KOffice
- Lotus eSuite
- Quadratron Cliq

- TeX

The final section presents two entirely different approaches to desktop computing:

- The VMware virtual platform, which lets you run Windows and Linux simultaneously on a single computer
- WINE, which lets you run DOS and Windows programs under Linux

8.1.1 Applix Applixware

Applix distributes Applixware, a desktop suite available for a variety of platforms, including Microsoft Windows 9x, Microsoft Windows NT, and several varieties of UNIX, including Linux. Applixware is commercial software, but much less expensive than Microsoft Word. Currently, Applixware for Linux is priced at \$99 (US); you can purchase it from the Applix web site, <http://www.applix.com/>. Print documentation and support programs are also available, at additional cost. If you prefer to try before buying, you can download a feature-disabled demo version; however, the download is quite large (over 38 MB) and therefore time-consuming (over 3 hours at 28.8 Kbps) unless you have a high-speed connection to the Internet.

Applixware includes eight main components:

- Words, a WYSIWYG word processor
- Spreadsheets, a graphical spreadsheet application
- Presents, a presentation graphics application
- Graphics, a drawing, charting, and graphics editing application
- Mail, an email client
- HTML Author, an application for authoring web documents

- Data, an application for accessing data held in Oracle, Informix, Sybase, or CA-Ingres databases
- Builder, an application development environment for decision support systems

8.1.1.1 Running Applixware

Once you've installed Applixware, you can run it by starting X and using a terminal window to issue the following commands:

```
DISPLAY=localhost:0  
/opt/applix/applix
```

If you installed Applixware to a directory other than */opt/applix*, you'll need to adapt the second command accordingly.

When Applixware starts up, it displays a window containing the Applixware main menu, shown in [Figure 8.1](#). From the toolbar of the main menu, you can simply click to launch Words, Graphics and Presents, Spreadsheets, Mail, or Data. Alternatively, you can click on the asterisk icon, causing Applixware to display a menu, as shown in [Figure 8.2](#).

Figure 8.1: The Applixware main menu

 Figure 8.1

Figure 8.2: The Applixware application menu

 Figure 8.2

[Figure 8.3](#) shows the Words application. Words includes a spelling checker and supports document templates, tables, multimedia, and live links to Applixware and third-party documents. Moreover, it lets you import and export a variety of document formats, including Microsoft Word 2.0, 6.0, 7.0/95, and 97.

Figure 8.3: The Words application

 Figure 8.3

Figure 8.4 shows the Presents application, which lets you create presentations in a way that resembles that provided by Microsoft PowerPoint. Presents provides features such as transitions and animation, HTML export, and templates. Its drawing tools let you draw lines, curves, and shapes, and perform other operations, such as rotations and fills. Presents can also import and export PowerPoint presentations and a variety of other document types.

Although a Linux application, such as Words or Presents, may be able to import a Microsoft Office document, the application may not be able to handle the full range of document and object types that can be embedded in an Office document. Some document and object types, for example, require access to a host application that runs only under Microsoft Windows.

Moreover, some Linux applications do not currently support all the styles and options of their Office counterparts. For example, if a document uses Microsoft's Visual Basic for Applications, the document will probably not function correctly under Linux.

If you plan on moving documents to and from Office, you may want to save your documents in a portable format (such as RTF), rather than a Microsoft proprietary format. This will significantly improve the chances that your document will work correctly both in Windows and Linux.

Figure 8.4: The Presents application

 Figure 8.4

Figure 8.5 shows the Spreadsheets application, which provides functions similar to those of Microsoft Excel. Spreadsheets lets you create ordinary spreadsheets as well as 3D charts and sheets, and

supports live links to objects created by Applixware and third-party applications, including Applix Data, which provides access to relational databases. Spreadsheets implements Applix's Extension Language Facility (ELF), which lets you add custom functions to the hundreds of built-in functions. Spreadsheets can import and export a variety of document types, including Lotus 1-2-3 (WKS, WK1, WK3, and WK4 files) and Microsoft Excel 3.0, 4.0, 5.0, 7.0/95, and 97.

Figure 8.5: The Spreadsheets application

 Figure 8.5

8.1.2 StarOffice

Another popular Linux desktop suite is StarOffice, a product of Sun Microsystems. StarOffice is available free of charge from Sun's web site, <http://www.sun.com/products/staroffice/>. The download is about 70 MB, requiring over 6 hours at 28.8 Kbps. The office suite is also available on CD-ROM, for a cost of \$9.95 plus shipping and handling.

StarOffice includes an extensive range of applications, including:

- StarDesktop, a file manager and desktop
- StarWriter, a word processor
- StarCalc, a spreadsheet application
- StarDraw, a drawing application
- StarImpress, a presentation graphics application
- StarBase, a relational database
- StarSchedule, a personal organizer that keeps track of events, tasks, contacts, and projects
- StarMail, an email client

- StarDiscussion, an Internet newsgroup reader
- SharChart, a business graphics application
- StarImage, an image editor that provides many special effects
- StarMath, a formula editor

8.1.2.1 Installing StarOffice

To install StarOffice from a StarOffice CD-ROM, log in, mount the CD-ROM, change to the directory that is the mount point of the CD-ROM device, change to the *Office50/english/prod_lnx* subdirectory, and issue the `setup` command:

```
su -
mount -t iso9660 /dev/cdrom /cdrom
cd /cdrom/Office50/english/prod_lnx
./setup
```

The install program will guide you through the installation process.

If you have downloaded the personal edition, you have a tar file, rather than a CD-ROM diskette. The tar file has a name similar to *so501_01.tar*. Change to the directory in which you saved the tar file (for example, */download*) and issue the following commands:

```
tar xvf so501_01.tar
cd so501_inst
./setup
```

If you saved your tar file using a name other than *so501_01.tar*, you'll need to adapt the first and second commands accordingly. As when you install from CD-ROM diskette, the install program will guide you through the installation process.

Whether you're installing from CD-ROM diskette or a downloaded file, you should check the *README* file in the *Office50* directory for

additional information and instructions that will help you install and use StarOffice.

8.1.2.2 Running StarOffice

Once you have installed StarOffice, you can run it by starting X and using a terminal window to issue the following commands:

```
PATH=$PATH:/opt/Office50/bin  
soffice
```

If you installed StarOffice into a directory other than */opt/Office50*, you'll need to adapt the first command accordingly.

When StarOffice loads, it automatically launches the StarDesktop application, as shown in [Figure 8.6](#). StarDesktop resembles Windows Explorer and the GNOME File Manager. It lets you point, click, drag, and drop to open and manage document files and folders.

Figure 8.6: StarDesktop

 Figure 8.6

[Figure 8.7](#) shows StarWriter, StarOffice's word processor. StarWriter provides PGP-encrypted email and other features not found in Microsoft Office 97. Like Applixware's Words, it imports and exports a variety of document types, including Microsoft Word 97.

Figure 8.7: StarWriter

 Figure 8.7

[Figure 8.8](#) shows StarImpress, the presentation graphics application included in StarOffice. Like Applixware's Presents, StarImpress imports and exports a variety of document types, including PowerPoint 97. StarImpress provides templates, animation and transitions, 3D effects, charts, and many other features.

Figure 8.8: StarImpress

 Figure 8.8

Figure 8.9 shows StarDraw, StarOffice's drawing program. StarDraw provides a wide range of drawing tools and effects, and works with a variety of file types, including even Windows Metafile (WMF).

Figure 8.9: StarDraw

 Figure 8.9

In addition to the bread-and-butter applications provided by Applixware, StarOffice provides several smaller, but quite useful, applications. For example, Figure 8.10 shows StarSchedule, which can help you keep track of the tasks on your to do list.

Figure 8.10: StarSchedule

 Figure 8.10

8.1.3 Corel WordPerfect for Linux

Another popular desktop application is Corel's WordPerfect for Linux, available free of charge for personal use from Corel's web site, <http://www.corel.com/>. WordPerfect is more a word processor than a complete desktop suite; for example, no spreadsheet application accompanies it. However, WordPerfect does provide many functions and features, including:

- a file manager
- drawing and charting
- spreadsheet functions in tables
- HTML authoring

A personal retail version (currently available for \$69.95 US) adds features such as:

- advanced drawing and charting functions
- version control
- clipart images, photos, textures, templates, and fonts

The downloadable version is about 24 MB in size, about a 2 hour download at 28.8 Kbps.

8.1.3.1 Installing WordPerfect for Linux

To install the downloaded version of WordPerfect for Linux, change to the directory that contains the downloaded file and issue the following command:

```
DISPLAY=localhost:0
```

Then start X, open a terminal window, and issue the following commands:

```
cd /download
tar zxvf wp.tgz
./Runme
```

If your downloaded file has a name other than *wp.tgz* or if it resides in a directory other than */download*, you'll need to adapt these commands accordingly.

The install program will guide you through the installation process.

To install the CD-ROM version of WordPerfect for Linux, follow the instructions that accompany the CD-ROM.

Some web sites, including the Corel web site, may provide WordPerfect for Linux as an uncompressed file. If the `tar` command

fails with an error message indicating that the file is not in the gzip format, use the following command to unpack the file:

```
tar xvf wp.tgz
```

8.1.3.2 Running WordPerfect for Linux

Once you've installed WordPerfect for Linux, you can run it by starting X and using a terminal window to issue the command:

```
/opt/corelwp8/wpbin/xwp
```

[Figure 8.11](#) shows WordPerfect for Linux in operation.

Figure 8.11: WordPerfect for Linux

 Figure 8.11

8.1.4 Other Desktop Applications

[Table 8.1](#) describes some other desktop applications available for Linux. Notable among these is TeX, a freely available desktop publishing package of unsurpassed power. Because of its powerful and sophisticated handling of formulas and mathematical notation, TeX is a favorite among scientists and mathematicians.

Table 8.1: Other Desktop Applications for Linux

Application	Web site	Description

Application	Web site	Description
Angoss SmartWare	http://www.angoss.com/	Includes a word processor, spreadsheet, and relational database. Print- and save-disabled demo version available for free download. Registration fee \$50 (US).
Axene Office	http://www.axene.com/	Includes a word processor, desktop publishing application, and spreadsheet. Linux license \$49 (US). Printed manuals \$60 (US) per application.
KOffice	http://koffice.kde.org/	GPLed desktop suite. Currently in alpha release.
Lotus eSuite	http://www.lotus.com/home.nsf/tabs/esuite1	Java applets that provide a word processor, spreadsheet, presentation graphics, project scheduler, chart, SQL/JDBC, and CGI gateway. Single-user license \$127 (US).
Quadratron Cliq	http://www.quad.com/	Character-based desktop suite. Single-user license \$49 (US).

Application	Web site	Description
TeX	http://www.ctan.org/	GPLed desktop publishing application. Sophisticated and powerful but somewhat cumbersome to use.

8.2 Other Approaches to Desktop Computing

If, after trying various desktop suites for Linux, you find yourself pining for your original Windows desktop suite, you can possibly coax your desktop suite to cooperate with Linux in either of two ways:

- the VMware virtual platform
- WINE

This section of the chapter explains these remaining options.

8.2.1 The VMware Virtual Platform

VMware, Inc. recently released a beta version of a product it calls VMware for Linux. The company promises a commercial release of the product that should be available by the time you read this. A related product, VMware for Windows NT, should also be available. For the latest information, check VMware's web site, <http://www.vmware.com/>.

As the names suggest, VMware for Linux runs under Linux whereas VMware for Windows NT runs under Microsoft Windows NT. Each product lets you run a so-called guest operating system alongside the host operating system. Supported guest operating systems include:

- FreeBSD
- Linux
- MS-DOS 6
- Solaris 7 Intel Edition
- Windows 2000 Professional Beta

- Windows 3.1
- Windows 95
- Windows 98
- Windows NT 4.0

For example, using VMware, your Linux system can launch and execute your favorite Windows 9x applications, including desktop suites. Preliminary experience with VMware suggests that it is a robust and efficient means of running legacy desktop applications under Linux.

A time-limited demo version of VMware for Linux is available on the company's web site. List price for both versions of VMware is \$299 (US); once the product is released, it can be purchased from the company's web site. For VMware to operate, you must have copies of the guest operating system and any desired applications on the hard drive of your Linux system.

8.2.2 WINE

WINE (a recursive acronym for "WINE is not an emulator") takes a different approach to supporting legacy applications. WINE is a GPLed Unix implementation of the Windows 3.x and Win32 application programming interfaces (APIs). By installing WINE on your system, you can run MS-DOS and Windows applications under Linux. Because WINE implements the APIs themselves, you need not have a copy of MS-DOS or Windows on the hard drive of your Linux system.

The WINE project is an ongoing effort. Many MS-DOS and Windows applications run well under WINE, but others use functions that are as yet only partially implemented. The WINE Development HQ web site, <http://www.winehq.com/>, provides access to a database that records user evaluations of WINE's ability to run various

applications. At the time of writing, the database contained 127 reports of applications scoring a perfect rating of 5.

8.2.2.1 Getting and installing WINE

WINE is frequently updated and improved and is not yet complete. Therefore, rather than installing WINE from this book's CD-ROM, you should obtain the latest version of WINE <http://packages.debian.org/wine>. To install WINE, issue the commands given in **Appendix C, *The Debian Package Management Utilities***.

8.2.2.2 Running WINE

If your Linux partition mounts your Windows partition (if any) as `/c`, you're ready to run WINE. Otherwise, you must first establish a simple directory structure. The easiest way to do so is to log in as root and issue the following commands:

```
mkdir -p /c/windows/system  
> /c/windows/win.ini
```

WARNING: If your Linux system mounts your Windows partition as `/c`, these commands will damage your Windows configuration. Do not issue them; you don't require them in order to run WINE.

To prepare to run an MS-DOS or Windows application, copy the application and any necessary DLLs or other files to the `/c/windows` directory. For example, to prepare to run the Minesweeper program, copy `winmine.exe` and `winmine.ini` to `/c/windows`. To run the application, you must have started X. Simply issue the `wine` command, specifying the application as an argument. For example, to run the Minesweeper program, issue the command:

```
wine winmine.exe
```

Figure 8.12 shows the Minesweeper program running under WINE.

Figure 8.12: Minesweeper running under WINE

 Figure 8.12



Learning Debian GNU/Linux

By Bill McCarty

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

9. Playing Linux Games

Contents:

[A Survey of Linux Games](#)

[Closeups of Some Popular Games](#)

In the last chapter you learned how to use Linux to help you work; in this chapter you'll learn how to use Linux to help you play. A variety of challenging and exciting games is available for Linux; many of them are free. In addition, you can use WINE to run a variety of commercial games originally written for Microsoft Windows.

9.1 A Survey of Linux Games

Linux includes several popular games. In addition, many Linux games are available on the Web. [Table 9.1](#) lists some of the most popular sites offering Linux games and [Table 9.2](#) describes some of the most popular Linux games. However, you'll find almost every game you could want on the Debian web site, <http://www.debian.org/>.

Table 9.1: Popular Linux Game Web Sites

Web Site	URL
Freshmeat	http://freshmeat.net/
Linux Game Tome	http://happypenguin.org/
Linux Games Page	http://www.linuxgames.com/
Linux Quake Page	http://www.planetquake.com/linux/
linuxquake.com	http://www.linuxquake.com/
Loki Games	http://www.lokigames.com/
Slashdot	http://www.slashdot.org/

Table 9.2: Some Popular Linux Games

Game	Type	Description
BZFlag	Action	Multi-player tank battle game.
Civilization: Call To Power	Strategy	A commercial Linux version of the sequel to Micropose's Civilization. Supports network play.
CrossFire	Role Playing	Resembles Rogue. Supports multi-player network play.
DOOM	Action	Classic action game. Requires <i>doom.wad</i> file from licensed copy of game.
Freeciv	Strategy	Resembles Microprose's Civilization. Supports network play.
Illust Logic	Puzzle	A paint-by-numbers puzzle, wherein you strive to paint cells of a canvas.
Koules	Arcade	Multi-player action game. Supports console or X11 play.
NetHack	Role Playing	A quest game resembling Rogue. Supports console or X11 play.

Game	Type	Description
Netrek	Arcade	Multi-player 2-D battle simulation with a Star Trek theme.
PySol	Card	Twenty different versions of solitaire.
Quake	Action	A Linux version of the commercial game Quake. Requires the PAK file from the game CD-ROM diskette. Supports console or X11 play.
Quake II	Action	In the opinion of many, the definitive multi-player 3D action shoot 'em up. Requires license fee. Supports console or X11 play.
Snes9X	Utility	A portable, freeware emulator of the Super Nintendo Entertainment System (SNES). Allows you to run Nintendo64 games on a PC.
Starcraft Linux- installation HOWTO	Document	Describes the procedures for installing and running the commercial game Starcraft on a Linux system using WINE.
XBomber	Arcade	Resembles the classic Bomberman game.

Game	Type	Description
Xmame	Arcade	Emulator for popular arcade games. Requires ROM image from the original game.
XPat2	Card	An assortment of solitaire games.
XShipWars	Action	A graphical MUD (multi-user dungeon) with a space exploration theme.

As you can see, many types of games are available, including action games, arcade games, card games, puzzles, role playing games, and strategy games. Some Linux games can be played from the console; others require X. Many Linux games let multiple players compete at separate computers connected via a network, such as the Internet. Linux games may be freeware, shareware, or commercial software. In addition, Linux software such as Snes9X and WINE lets you play games originally written for systems other than Linux.

9.2 Closeups of Some Popular Games

In this section, you'll get acquainted with three popular games you can run under Linux:

- DOOM, which runs on a console or under X
- Quake II, which runs on a console or under X
- StarCraft, which runs under X by using WINE

9.2.1 DOOM

Originally written for MS-DOS by id Software (<http://www.idsoftware.com/>), DOOM is the archetypal 3D action game. You play the role of a space marine, fighting your way through a series of bases on the moons of Mars that have been invaded by aliens. The game features real-time 3D graphics and stereo sound effects. [Figure 9.1](#) shows a typical game screen.

Figure 9.1: A typical DOOM screen



If you prefer more cunning opponents, you can play the game in network mode, via modem, or an IPX or TCP/IP network. Network mode lets you match wits with two to four human opponents.

To run DOOM in a console, type the following command:

```
sdoom -warp 1 1
```

If you prefer to run DOOM under X, type the following command:

```
xdoom -warp 1 1
```

To call up DOOM's main menu, simply press **Esc**.

You can learn more about DOOM by visiting the DOOM Archives at <http://www.idsoftware.com/archives/doomarc.html> and the Linux DOOM FAQ at <http://jcomm.uoregon.edu/~stevev/Linux-DOOM-FAQ.html>. You might also enjoy lxDoom, a Linux port of Boom, which is an enhanced version of DOOM.

9.2.2 Quake II

Like DOOM, Quake II was also written by id Software. However, Quake II is a much more modern and sophisticated program than DOOM. For example, Quake II's multi-player mode lets as many as 32 players wander the planet of Stroggos. And, the single-player mode pits you against 18 artificially intelligent adversaries, who dodge your careless shots with agile ease. [Figure 9.2](#) shows a typical Quake II game screen.

Figure 9.2: A typical Quake II screen



To run Quake II under X, change the working directory to the directory in which you installed Quake II, and issue the command:

```
./quake2 +set vid_ref softx
```

To run Quake II using a virtual console, issue the command:

```
./quake2 +set vid_ref soft
```

Once the game is running, you can access its main menu by pressing **Esc**.

9.2.3 Starcraft

Starcraft is a real-time strategy game published by Blizzard Entertainment, in which you participate as the leader of a group of humans exiled at the edge of galactic space. Your objective is to

assemble a military force capable of dominating two other species, the Protoss and the Zerg. To do so, you must build bases, vehicles, and weapons, and train soldiers for combat.

Starcraft was not written for Linux; however, you can run it under Linux by using WINE. Assuming that you have WINE installed on your system, the following sections explain how to install and run Starcraft:

9.2.3.1 Installing Starcraft

If you have the commercial version of Starcraft, mount the CD-ROM diskette and locate the file *install.exe*. If you want to try the Starcraft demo, download the file *scdemo.exe*, which is a little over 28 MB in size. You can find the file on Blizzard's Web site (<http://www.blizzard.com/>) and elsewhere.

Change the current directory to the directory containing *install.exe* or *scdemo.exe* and use WINE to install Starcraft:

```
wine -display localhost:0 -winver win95 scdemo.exe
```

When asked if you want to install DirectX 5, highlight the No, But Continue option and click on OK. Choose the directory in which you want to install Starcraft (the default choice is generally acceptable) and click on OK. If the program asks if you want to register via the Internet, respond by clicking on No. Finally, click on Exit to terminate the install program.

Under Linux, Starcraft operates in 256-color 640×480 mode. Therefore, you must adjust your X configuration to provide this mode. Here's a quick way to do that. First copy the file */etc/X11/XF86Config* by issuing the following commands:

```
cd /etc/X11
cp XF86Config XF86Config.SAVE
cp XF86Config XF86Config.StarCraft
```

Now, edit the file *XF86Config.StarCraft* using the editor of your choice. Find the "Screen" section that specifies the X server (driver) you use and change the Depth parameter to 8 and the Modes parameter to "640x480". Delete any additional modes that appear. When you're done, the screen section should resemble the following:

```
Section "Screen"
    Driver      "svga"
    Device      "Millennium"
    Monitor     "Viewsonic17GS"
    Subsection  "Display"
        Depth   8
        Modes   "640x480"
        ViewPort 0 0
        Virtual 640 480
    EndSubsection
EndSection
```

9.2.3.2 Playing Starcraft

To play Starcraft, replace your *XF86Config* file with the newly edited one:

```
cp XF86Config.StarCraft XF86Config
```

Then start X:

```
startx
```

Launch an X terminal, make the Starcraft installation directory the current directory, and start the Starcraft program:

```
#
cd "/c/Program Files/Starcraft Shareware(ED)"
#
wine -display localhost:0 -winver win95 -depth 8 \
>
-geom 640x480 Starcraft.exe
```

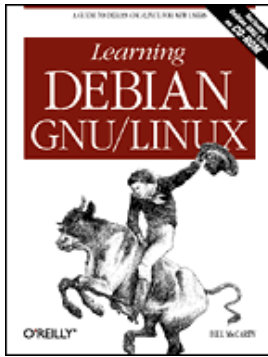
If you installed Starcraft to a directory other than */c/Program Files/Starcraft Shareware(ED)*, you should adjust the commands accordingly.

Now, play Starcraft until you completely dominate the galaxy. The tutorial mission will help you learn how to do so. Of course, planning and executing a winning strategy will require practice.

9.2.3.3 Ending a Starcraft session

When you're done, press F10 to obtain a game menu, exit the current mission, and exit the game. Then, replace your original *XF86Config* file by entering the commands:

```
cd /etc/X11
cp XF86Config.SAVE XF86Config
```

Learning Debian GNU/Linux

By Bill McCarty

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

10. Setting Up a Linux-Based LAN

Contents:

[Introduction](#)

[Network Administration](#)

[Samba](#)

[Samba Client Configuration and Use](#)

This chapter explains how to set up a local-area network (LAN) that includes a Linux Samba server, which lets Microsoft Windows and UNIX systems access shared files and printers hosted by your Linux system. The chapter explains how to administer a simple LAN and describes how to install, configure, and administer Samba servers and clients. Integrating your Linux system with an existing LAN is no more complicated than setting up your own LAN; the chapter also explains how to connect to an existing network. The chapter also explains how to use Linux backup and recovery utilities so that client systems can create and use backups stored on the server.

10.1 Introduction

One of the great strengths of Linux is its powerful and robust networking capabilities. The good news is that everything about Linux's networking setup is open to inspection and completely configurable. Nothing is hidden from the user, and no parameters are forced on you. The challenge is to get the most out of this setup for your needs.

Basic networking principles don't differ much between Windows and Linux, and indeed the principles aren't unfamiliar. This chapter starts with an overview of networking, and then looks in more detail at Linux networking on a Local Area Network (LAN). In the next two chapters, you'll learn about making a dialup Internet connection, and setting up Wide Area Network (WAN) services.

Most computers today handle network traffic much as the post office handles mail. Think, for example, of the steps involved in sending and receiving a letter. Your postal carrier must know where to drop off and where to pick up mail. So your home must have some kind of recognizable *interface*; we call this a mailbox. And whereas your postal carrier may know your neighborhood quite well, delivery in other areas will require other carriers. Mail is passed to these other carriers through a *gateway*; we call this the Post Office. Although you can think of the whole postal system as one big network, it's easier to understand if you think of it as a hierarchy of *subnetworks* (or *subnets*): the postal system is divided into states, states are divided by zip code, zip codes contain a number of streets, and each street contains unique addresses.

Computer networking mirrors this model. Let's trace an email message from you to a coworker. You compose the message and press Send. Your computer passes the message to a network interface. This interface may be a modem by which you dial up an ISP, or it may be an Ethernet card that connects you to a LAN. Either way, on the other side of the interface is a gateway machine. The gateway knows how to look at the address of the recipient on the email message, and interpret that message in terms of networks and subnets. Using this information, your gateway passes the message to other gateways until the message reaches the gateway for the destination machine. That gateway in turn delivers the message via a

recognizable interface (such as modem or Ethernet card) to the recipient's inbox.

If you review this story, you can easily see what parts of networking you'll need to configure on your Linux system. You'll need to know the address of your machine. Just as the town name *Menlo Park* and the zip code *94025* are two different names for the same location, you may have both a name, called a *hostname*, and a number, called an *IP number*, that serve as the address for your machine.

To translate between these two notations, you may need to know the address of a *Domain Name Server*. This is a machine that matches IP numbers with hostnames. You'll also need to know the address of a gateway machine through which network traffic will be routed. Finally, you'll need to be able to bring up an interface on your system for networking, and you'll need to assign a route from that interface to the gateway.

While all of this can seem complex, it really isn't any more complex than the postal system, and functions in much the same way. Fortunately, Linux comes with tools to help you automate network configuration. In this chapter you'll look at networking on a LAN, and we'll start by looking at how to set up LAN networking.

10.2 Network Administration

The Debian GNU/Linux install program lets you specify a network configuration that's used when your system is first booted. If your network configuration changes, you can re-install Linux. However, you can spare yourself much inconvenience by learning how Linux stores its network configuration. As you'll see, by using a text editor to revise some files, you can alter your system's network configuration without going through the pain of re-installing Linux.

10.2.1 Network Hardware Configuration

If you replace your network adapter card with a different model card, you must run the `modconf` program, which lets you specify the driver that operates your card. To do so, simply login as `root` and type the command:

```
modconf
```

You're already familiar with the `modconf` program. It's the same program you used to specify drivers when you originally installed Linux. If you have difficulty using `modconf`, refer to the section titled **Section 3.1.2.13, "Configuring device driver modules"** in **Chapter 3**,

Installing Linux. You must reboot your system before changes made by `modconf` take effect.

10.2.2 Basic Host Information

When you installed Linux, you specified a hostname for your system. If you want to change the hostname associated with your system, you can edit the file `/etc/hostname` by using `ae` or another editor of your choosing. Because the file - like most configuration files - has restrictive permissions, you must login as `root` in order to modify it.

The format of the `/etc/hostname` file is simple. The file contains a single line, which contains the hostname of your system; for example, `debian`. If you change the hostname, be sure to specify only the hostname itself; do not specify a fully qualified hostname that includes the domain name (for example, `debian.ora.com`).

10.2.3 Name Server Specification

When you installed Linux, you may have specified one or more nameservers. Your system accesses a nameserver when it needs to determine the network address that corresponds to a hostname. If your network configuration changes, you may need to specify a new nameserver or servers. Your ISP should provide you with the proper IP address or addresses.

The network addresses of your system's name servers are specified in the file `/etc/resolv.conf`, which you can edit by using `ae` or another editor while logged in as `root`. The format of the file is simple, though not as simple as that of the `/etc/hostname` file. To specify a name server, include a line of the form:

```
nameserver  
xxx.xxx.xxx.xxx
```

where `xxx.xxx.xxx.xxx` denotes the network address (IP number) of the name server; for example, `192.168.1.1`. You can include as many as three such lines; when your system needs to determine a network address, it will attempt to contact the name server specified by the first such line. If that server is unavailable, your system will attempt to contact the name server

specified in the second such line, if any. If that werver is unavailable, your system will contact the name server specified in the third such line, if any.

10.2.4 Routing and Gateways

If your computer is part of a local area network attached to the Internet, your computer doesn't generally send data packets directly to Internet hosts. Instead, it sends data packets to a designated computer - called the *gateway* - on the local area network. The gateway forwards data packets to the Internet on behalf of your system. It also performs the complementary service, forwarding data packets from Internet hosts to your system.

NOTE: If your system connects to the Internet via PPP, the PPP system establishes a network configuration dynamically. You'll learn how this works in the next chapter.

The information that describes your local area network is contained in the file */etc/init.d/network*, which you can easily edit. Here's a typical */etc/init.d/network* file:

```
#!/bin/sh
inconfig lo 127.0.0.1
route add -net 127.0.0.0
IPADDR=192.168.1.10
NETMASK=255.255.255.0
BROADCAST=192.168.1.255
GATEWAY=192.168.1.1
ifconfig eth0 ${IPADDR} netmask ${NETMASK} broadcast ${BROADCAST}
route add default gw ${GATEWAY} metric 1
```

The lines you're concerned with are the lines four through eight, each of which has the following form:

```
variable=  
ipnumber
```

The lines associate a name, given by *variable*, with a network address, given by *ipnumber*. The variables are referenced by the following three lines, which are commands that configure networking.

To change your network configuration, you need merely to associate the proper IP number with each variable. You can do so by logging in as *root*

and modifying the `/etc/init.d/network` file by using a text editor. [Table 10.1](#) describes each variable. Your network administrator should be able to provide you with the proper values.

Table 10.1: Network Configuration Variables

Variable	Meaning
IPADDR	Specifies the network address of your system.
NETMASK	Specifies the network address of your network, by indicating which bits of the 32-bit network address of your system pertain to the network and which pertain to the system. Many local area networks are so-called Class C networks, which require a netmask of 255.255.255.0.
BROADCAST	Specifies the address used to send a message to every system on the local area network. Often you can determine the broadcast address of a local area network from the address of a system on the local area network: simply replace the last of the four components of the network address of the host by 255.
GATEWAY	Specifies the network address of the gateway used by your system.

10.2.5 Hostname Search Path

Your Linux system can use as many as three methods to determine the IP address that corresponds to a hostname. Your system can:

- Query a DNS server (you configured your system's DNS client earlier)
- Read the contents of the file `/etc/hosts`, known as the *hosts file*, which you'll learn about in the next subsection

- Query an NIS (Network Information System) server

However, unless your system is part of a sophisticated local area network, it's unlikely that an NIS server is available. Therefore, most systems can query a DNS server and, failing to obtain an answer, read the */etc/hosts* file. Alternatively, most systems can read the *etc/hosts* file and, failing to obtain an answer, query a DNS server. The second alternative is generally better, because reading the */etc/hosts* file takes less time than querying a DNS server.

The */etc/host.conf/* file specifies which of these three operations are performed, and the order in which they're attempted. You can edit this file by logging in as `root`. Here's a typical file:

```
order hosts,bind
multi on
```

The `order` line specifies that the system should first consult the */etc/hosts* file and then query a DNS server, referred to as `bind` because of the Berkeley Internet Name Daemon, an early DNS server.

The `multi` line specifies that your system will attempt to locate all possible names for a host when it reads the */etc/hosts* file. Unless that file is very large (hundreds or thousands of lines), you should include the `multi` line.

10.2.6 Miscellaneous Network Configuration Options

The hosts file, */etc/hosts*, lets your system determine the network address number that corresponds to a hostname, without querying a DNS server. Besides being faster than querying a DNS server, the */etc/hosts* file is always available.

Entries in the file have two parts:

- an IP address
- a hostname, or a list of hostnames separated by spaces

By default, the hosts file contains an entry that associates the hostname localhost with the IP address 127.0.0.1. It's not necessary that you include any other entries in the */etc/hosts* file. However, most system administrators

include at least a second line, which associates the local hostname with its network address. Here's a typical file:

```
127.0.0.1      localhost
192.168.1.10   debian.mccarty.org  debian
```

Notice that the second line gives both the fully qualified hostname, consisting of the hostname and domain name, as well as the hostname alone.

The */etc/networks* file, known as the networks file, performs a function similar to that of the hosts file; whereas the hosts file associates hostnames with network addresses, the networks file associates networks' names with network addresses. By default, the networks file contains a single line associating the network address of the local area network with the name localnet:

```
localnet 192.168.1.0
```

Generally, it's not necessary that you add other entries to the networks file. However, by doing so, you can access frequently used networks by name even if your DNS server is unavailable.

10.3 Samba

Available since version 3.11 of Microsoft Windows, printer and file sharing are two of Window's most useful features. For example, outfitting each computer in a large office with a laser printer would be quite expensive. But printer sharing lets you reduce the cost of providing every user with printing capability; with printer sharing, each computer system in the office can print to a single printer.

Just as printer sharing lets your computer access a printer attached to another computer, file sharing lets your computer access files stored on another computer. File sharing makes it quick and easy to transfer data from one system to another and avoids the confusion that results when everyone has their own - possibly out of date or inconsistent - copy of important data files they could not otherwise access.

To provide printer and file sharing, Microsoft Windows uses a facility known as SMB (Server Message Block). This same facility is sometimes known as NetBIOS or LanManager. Thanks to Andrew Tridgell and others, Linux systems provide support for SMB via a package known as Samba. Like SMB, Samba lets you:

- Share printers and files among Microsoft Windows, OS/2, Netware, and Unix systems
- Establish a simple nameserver for identifying systems on your local area network
- Backup PC files to a Linux system and restore them
- Administer users and passwords

Samba has proven its reliability and high performance in many organizations. According to the online survey at <http://www.samba.org/pub/samba/survey/ssstats.html>, Bank of America is using Samba in a configuration that includes about 15,000 clients, and Hewlett-Packard is using Samba in a configuration that includes about 7,000 clients.

10.3.1 Samba Server Installation

If you've never installed and configured a network server, Samba's a good place to begin; its installation and configuration are generally simple and straightforward.

Samba includes the `smbd` program, which runs as a daemon, several utility programs, man pages and other documentation, and the configuration file: `/etc/smb.conf`. You'll learn how to configure the `/etc/smb.conf` file in the next subsection.

NOTE: The pending unstable version of Samba places its configuration files in `/etc/samba`, rather than `/etc`. If you install a recent version of Samba, you should look in `/etc/samba` for the Samba configuration files.

10.3.2 Configuring Samba

The `/etc/smb.conf` file lets you specify a variety of options that control Samba's operation. The install script for Samba establishes a simple `/etc/sbm.conf` that may meet your requirements.

You can edit the `/etc/sbm.conf` file to suit your special needs by using your favorite text editor. However, Samba includes a tool called `swat` that lets you view and change options by using your Web browser, which is generally much easier than using a text editor. The `swat` tool verifies the values of parameters you enter and provides online help. To access `swat`, point your browser to port 901 of your system. For example, you can use the URL <http://localhost:901/>. Your web browser will prompt you for a userid and password; specify `root` as the userid and give the appropriate password. [Figure 10.1](#) shows `swat`'s main menu, accessed by using the system's IP address in place of its hostname.

Figure 10.1: The Samba main menu

 Figure 10.1

To configure your Samba server, you click on simulated tool bar entries:

- Globals lets you configure global Samba variables (options)
- Shares lets you configure file shares

- Printers lets you configure shared printers
- Status lets you view the status of the Samba server
- View lets you view the *smb.conf* file
- Password lets you add and delete users and change user passwords

The main menu also provides convenient access to Samba documentation.

10.3.2.1 Configuring global variables

To configure global options, click on the Globals button on the tool bar. [Figure 10.2](#) shows the Global Variables page and [Table 10.2](#) describes the most important options. You can access additional options by clicking on Advanced View. To change an option, select or type the desired value. When you've changed all the options you want to change, click on Commit Changes, which causes the changes to take effect.

Figure 10.2: Samba global variables

 Figure 10.2

Table 10.2: Samba Global Variables

Option group	Option	Description
Base	workgroup	The workgroup name displayed when the server is queried by a client.
	netbios name	The name by which the server is known to the NetBIOS nameserver.
	server string	The text string displayed to describe the server.

Option group	Option	Description
Base (cont.)	interfaces	The IP address of the interface or the IP addresses of the interfaces through which Samba should listen. Each IP address is followed by a forward slash and a number that specifies the number of bits that pertain to the network portion of the IP address (usually 24). If this option is not set, Samba attempts to locate and automatically configure a primary interface.
Security	security	Specifies how Samba authenticates requests for access to shared resources. The default value, <code>user</code> , is helpful when the Samba server and its clients have many common userids. The value <code>share</code> can be useful when few common userids exist. The value <code>system</code> lets another SMB server perform authentication on behalf of the server. You should generally use the default value; see the Samba documentation for details.
	encrypt passwords	Specifies whether Samba will negotiate encrypted passwords, which are expected by Windows NT 4.0 SP3 and Windows 98.
	update encrypted	Allows automatic updating of an encrypted password when a user logs on using a non-encrypted password. This option is useful when migrating to encrypted passwords and should otherwise be set <code>off</code> .
	map to guest	Specifies Samba's action when a user attempts to log on using an invalid password. The <code>Bad User</code> option is generally appropriate.

Option group	Option	Description
	guest account	The Linux account used to provide services for guest users.
	hosts allow	A list of hosts that can access the server. If not specified, all hosts are permitted access.
	hosts deny	A list of hosts that cannot access the server.
Logging	log level	An integer that specifies the verbosity of log messages. A low value (such as 0) specifies that few messages are written to the log.
	log file	Specifies the name of Samba's log file.
	max log size	The maximum size of the log file in kilobytes (kb). When the specified size is exceeded, Samba begins a new log file. A value of zero lets the log file grow indefinitely large.
Tuning	read prediction	Specifies whether Samba will attempt to pre-read data from files, in order to speed data transfer. This code is disabled in Samba 2.0.
	socket options	Specifies TCP options that can improve performance. See the Samba documentation for details.

Option group	Option	Description
Printing	printcap name	Specifies the name of the <i>printcap</i> file used by the server.
	printing	Specifies how Samba interprets printer status information. Generally, <code>sysv</code> is an appropriate choice for a Linux system.
Logon	logon script	Specifies the path of a BAT file that is downloaded from the server and run when a user logs on to Samba.
	domain logons	Specifies whether Samba will serve Windows 9x domain logons for its workgroup. Note: Samba cannot yet serve Windows NT domain logons, which require a Primary Domain Controller (PDC).
Browse	os level	Specifies the level at which Samba advertises itself for browse elections. A high number makes it more likely that Samba will be selected as the browser. The value 65 will cause clients to prefer Samba to a Windows NT server.
	preferred master	Specifies whether the NetBIOS name server is the preferred master browser for its workgroup.
	local master	Specifies whether the NetBIOS name server will bid to become the local master browser on a subnet.

Option group	Option	Description
	domain master	Specifies collation of browse lists across a wide-area network (WAN). May result in strange behavior when a workgroup includes a Windows NT Primary Domain Controller (PDC).
WINS	wins server	Specifies the IP address of the WINS server with which the NetBIOS nameserver should register itself, if any.
	wins support	Specifies that the NetBIOS nameserver should act as a WINS server. Useful when the network includes several subnets. Do not specify this option for multiple systems of a single network.
Locking	strict locking	Specifies whether the server will automatically lock files and check locks when files are accessed. Enabling this option may slow performance.

You probably won't need to make many changes to Samba's global variables. Setting the `workgroup` and `netbios name` is sufficient for most users. If your system has more than one network adapter card, you'll also need to set the `interfaces` variable.

WARNING: If your system is attached to a cable modem or other interface that makes it available to other network users, you should consider specifying security options that prevent unauthorized users from accessing your files or printer. Like other Samba users, the author has had a rogue system (in Pakistan) repeatedly attempt to access his shared files. To ensure that such attempts are unsuccessful, you might (for example) use the `hosts allow` option to restrict the hosts allowed to access your Samba server.

10.3.2.2 Configuring file share parameters

To establish and maintain file shares, you use the Shares button on the tool bar. [Figure 10.3](#) shows the Share Parameters page.

Figure 10.3: Samba file share parameters



You can create a new share by typing its name and clicking on Create Share. To delete a share, choose it from the drop-down list, and click on Delete Share. To work with an existing share, choose it from the drop-down list, and click on Choose Share. When you click on Choose Share, the page shown in [Figure 10.4](#) appears. This page lets you view and change a variety of share options. [Table 10.3](#) describes the available share options. You can access additional options by clicking on Advanced View. As with the global options, you may not need to change many (if any) share options. Likely candidates for change are the `comment`, `path`, `read only`, and `create mask` options.

Figure 10.4: Samba file share parameters



Table 10.3: Samba File Share Options

Option Group	Option	Description
Base	<code>comment</code>	The description displayed when the file share is queried by a client.
	<code>path</code>	The path (directory or file) that is shared by the server.

Option Group	Option	Description
Security	guest account	The Linux account used to provide services for guest users.
	read only	Specifies whether access to the share is read-only.
	create mask	The default mode assigned to a newly created file within a shared directory.
	guest ok	Specifies whether guest access (access without a password) is allowed.
	hosts allow	A list of hosts that can access the file share. If not specified, all hosts are permitted access.
	hosts deny	A list of hosts that cannot access the file share.
Browse	browseable	Specifies whether the file share is visible in the list of shares made available by the server.
Locking	strict locking	Specifies whether the server will automatically lock files and check locks when files are accessed. Enabling this option may slow performance.

Option Group	Option	Description
Miscellaneous	available	Specifies whether the share is available; by setting this option to "no" you can prevent access to the share.
	volume	The volume label returned for the share.

10.3.2.3 Configuring printer share parameters

You configure printer share parameters in much the same way you configure file shares. Begin by clicking on the Printers tool bar button. The page shown in [Figure 10.5](#) appears. You can use the page to create a new printer share, delete a printer share, or modify an existing printer share.

Figure 10.5: Samba printer parameters

 Figure 10.5

If you select a printer from the drop-down list and click on Choose Printer, the page shown in [Figure 10.6](#) appears. [Table 10.4](#) describes the available print share options. You can access additional options by clicking on Advanced View. As with the global options and file share options, you may not need to change many (if any) printer share options. Likely candidates for change are the `comment`, `path`, `read only`, and `create mask` options.

Figure 10.6: Samba printer parameters

 Figure 10.6

Table 10.4: Samba Print Share Options

Option Group	Option	Description

Option Group	Option	Description
Base	comment	The description displayed when the printer share is queried by a client.
	path	The print spooling directory.
Security	guest account	The Linux account used to provide services for guest users.
	guest ok	Specifies whether guest access (access without a password) is allowed.
	hosts allow	A list of hosts that can access the printer share. If not specified, all hosts are permitted access.
	hosts deny	A list of hosts that cannot access the printer share.
Printing	print ok	Specifies whether printing is permitted. If this option is set to "no," clients may still be able to browse the printer share.

Option Group	Option	Description
Printing (cont.)	printing	Specifies the type of printer interface used, which determines what commands Samba issues to control the printer. "BSD" is generally a good choice.
	printer name	Specifies the name of the printer to which the printer share corresponds; "lp" is generally a good choice.
Browse	browseable	Specifies whether the printer share is visible in the list of shares made available by the server.
Miscellaneous	available	Specifies whether the printer share is available; by setting this option to "no" you can prevent access to the printer share.

10.3.3 Viewing Samba Server Status

The Status button on `swat`'s tool bar lets you view the status of the Samba server. [Figure 10.7](#) shows the page that appears when you click on Status. The page shows:

- the status of the server daemons (*smbd* and *nmbd*) and the version of Samba
- active connections
- active file and printer shares
- open files

Using the controls on the page, you can refresh the page contents, set the auto refresh interval, start and stop either daemon, or kill an active connection.

Figure 10.7: Samba status page

Figure 10.7

10.3.4 Viewing Samba Server Configuration

The View button on `swat`'s tool bar lets you view the Samba server's main configuration file, `/etc/smb.conf`. [Figure 10.8](#) shows the page that appears when you click on View. By default, the page shows only the basic configuration options; clicking on Full View causes `swat` to display every configuration option.

Figure 10.8: The Samba `/etc/smb.conf` file

Figure 10.8

10.3.5 Managing Users and Passwords

You can create userids for accessing Samba resources by clicking on `swat`'s Password tool bar button. [Figure 10.9](#) shows the Password page that appears.

Figure 10.9: The password page

Figure 10.9

The top part of the page, titled Server Password Management, lets you:

- Create a new userid
- Delete a userid
- Change the password associated with a userid
- Enable or disable a userid

The userids you specify using Server Password Management are those that your Samba server recognizes as authorized to access its resources.

The bottom part of the page, titled Client/Server Password Management, lets you change the password associated with a userid on a remote system running Samba or SMB. Changing a password by using Client/Server Password Management is often more convenient than logging in to the remote host and using its password change facility.

10.3.6 Troubleshooting Samba

Like any network server, Samba provides a wealth of options and facilities. If you thoroughly explore these facilities, you're likely to break your server. To avoid problems, you should keep a backup copy of your */etc/smb.conf* file. Doing so can be as easy as issuing the following command after Samba is up and running:

```
cp /etc/smb.conf /etc/smb.conf.SAVE
```

Then, if your server ceases to work, you can restore your old configuration by issuing the command:

```
cp /etc/smb.conf.SAVE /etc/smb.conf
```

You'll also need to restart your system (or at least the Samba daemons).

On the other hand, you may have difficulty in getting Samba to operate correctly in the first place. Or, you may need to change Samba's configuration and therefore be unwilling to simply restore its previous status.

In such cases, you can consult the documentation that accompanies Samba. In particular, peruse the file *DIAGNOSIS.txt*, which should be in the */usr/doc/Samba-2.0.3/docs/textdocs* directory, or its equivalent on your system. This file includes a step-by-step procedure for verifying the operation of your Samba server. When a step fails, you can consult the file to determine the likely causes and how to go about fixing the problem. Chances are, you'll be able to administer Samba without outside help; but, if you can't, you'll find the participants of the *comp.protocols.smb* newsgroup to be helpful.

NOTE: Some versions of the Samba package do not include the file DIAGNOSIS.txt. If your system lacks the file, you can obtain it from the Samba web site, <http://www.samba.org/>.

10.4 Samba Client Configuration and Use

Once you've got your Samba server up and running, you can access it via Microsoft Windows, Linux, and other operating systems. This section shows you how to do so and also how to use your Samba server to create backups of important data files on client systems.

10.4.1 Microsoft Windows Client

Microsoft Windows 3.11, 9x, and NT have built-in support for the SMB protocol, so systems running these operating systems can easily access your Samba server's resources. Under Microsoft Windows 9x and NT, you can access Samba resources by using the Windows Explorer. Log on using a userid that's authorized to access Samba resources. Then click on Network Neighborhood and you should see a subtree that corresponds to your Samba server. By expanding the subtree, you can see the browseable file and printer shares that are available. You can easily drag and drop files to and from a shared directory, assuming your userid is permitted the necessary access.

To use a shared printer, click on Start → Settings → Printers and then double click on Add Printer. The wizard will guide you through the setup procedure. Simply choose the Network Printer option and then browse to select the desired printer. If you configured the printer share without the browseable option, you cannot browse and therefore must type the name of the printer share. To do so, type two backslashes, followed by the name of your Samba server, followed by a single backslash, followed by the name of the printer share. For example, if you want to access a printer share named `lp` on the Samba server known as *SERVER*, you'd type `\\SERVER\lp`.

You can map a file share to a drive letter by using the Tools → Map Network Drive menu item of the Windows Explorer. Simply select an available drive letter and type the name of the file share, which consists of two backslashes, followed by the name of your Samba server, followed by a single backslash, followed by the name of the file share. For example, if you want to access a file share named `db` on the Samba server known as *SERVER*, you'd type `\\SERVER\db`.

If you have difficulty connecting to your Samba server, follow the procedure given in the preceding section on troubleshooting.

10.4.2 Other Clients

Of course, an SMB client is available for Linux; you'll learn about it in the next subsection. SMB clients are also available for most popular operating systems, including IBM OS/2 and Mac OS. You shouldn't expect to have trouble getting them to work with Samba. If your client seems not to work, simply follow the procedure given in the troubleshooting section.

10.4.3 Linux Client

The Samba packages include a simple SMB client that can access your Samba server and other SMB servers accessible to your system. To demonstrate that your client and server are working, log on using a userid that has Samba authorization and issue the following command:

```
smbclient -L localhost
```

You should see a list of the browseable shares available on your server. To query a different SMB server, issue the following command:

```
smbclient -L  
server
```

where *server* is the name of the SMB server you want to contact. Rather than log on using a authorized userid, you can explicitly specify a userid by using this command form:

```
smbclient -L  
server -U  
userid
```

To actually access resources via SMB, use the following command form:

```
smbclient -L  
service -U  
userid
```

where *service* specifies the name of the SMB host and share and *userid* specifies the userid to be used. The name of the SMB host should be preceded by two backslashes and followed by one backslash.

If the SMB server accepts your request, the client displays a special prompt:

```
smb:  
dir>
```

where *dir* indicates the current working directory on the SMB server.

To download a file from the server, issue the command:

```
get  
file
```

where *file* specifies the name of the file. To upload a file to the server, issue the command:

```
put  
file
```

where *file* specifies the name of the file. To list the contents of the current directory, issue the command:

```
dir
```

where *file* specifies the name of the file. To enter a subdirectory, issue the command:

```
cd  
dir
```

where *dir* specifies the name of the subdirectory. You can return to the parent directory by issuing the command:

```
cd ..
```

To exit the SMB client, issue the command `exit`. You can obtain a list of commands by issuing the command `help` or obtain help on a particular command by issuing the command:

```
help  
command:
```

where *command* specifies the command for which you want help.

Ivan Volosuk has written an X11 interface for `smbclient`, which you may find easier to use. You can learn more about it from his web page, available at <http://www.rt.mipt.ru/frtk/ivan/>.

You can use `smbprint` script included in the Samba package to print Linux files by using a printer share. However, you'll probably have to do some tweaking of configuration files and adjusting of shell scripts to get `smbprint` to work. This might be a good project for you to tackle after you've read **Chapter 13**,

Conquering the BASH Shell.

10.4.4 Using the Linux Samba Client for File Backup and Recovery

One of the most practical uses of the Linux SMB client is creating backup copies of files stored on a Microsoft Windows system. To do so, simply share the drive or directory containing the files you want to backup: using the Windows Explorer, right click on the drive or directory, click on Properties, click on the Sharing tab, and select the desired share options. Then, access the share from Linux using `smbclient`. Once you have the SMB prompt, move to the directory you want to backup, and issue the following SMB command:

```
tar c backup.tar
```

The files of the current directory and all its subdirectories will be backed up and stored in the file *backup.tar* on your Linux system. Of course, you can specify a filename other than *backup.tar*, if you wish. Once you've created the backup file, you can write it to a tape, a writable CD-ROM, or other media, if you need an offsite copy. If your backup requirements are meager, it may be sufficient merely to have a copy of the file on both your Windows system and your Linux system.

To restore a backup, move to the directory to which you want the files restored and issue the following SMB command:

```
tar x backup.tar
```

The SMB client restores each file from the tar file. Of course, you must have write access to the shared directory in order to be able to write the restored files.



Learning Debian GNU/Linux

By Bill McCarty

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

11. Getting Connected to the Internet

Contents:

[Connecting to the Internet](#)

[Configuring Your Modem](#)

[Using wvdial](#)

[PPP Client](#)

[Web Browser](#)

[gFTP FTP client](#)

[Using minicom and seyon](#)

[Making a PPP Connection Manually](#)

This chapter explains how to use Linux to connect to remote servers. First, it explains how to use `wvdial`, a program that makes it easy to connect to the Internet via a PPP connection provided by an ISP. Then, the chapter explains basic TCP/IP network concepts that you must know in order to administer a Linux system connected to the Internet or a local area network. So that you can use your knowledge of TCP/IP effectively, the chapter explains how to use `pppconfig` to configure and administer a system that connects to a local area network and to a remote server via PPP. Next, the chapter describes several popular network client applications available under Linux, including a web browser and an FTP client. The chapter then describes the use of `minicom` and `seyon`, which provide dial-out capabilities like those of Window's `hyperterminal`. Finally, the chapter shows how to make a PPP connection manually, by using `minicom`.

11.1 Connecting to the Internet

Most Internet service providers (ISPs) offer two primary types of service: shell accounts and PPP (point-to-point protocol) accounts. Shell accounts were more popular before the advent of the Web. A shell account lets you use your computer much as if it were a virtual console associated with a remote computer. You can type commands, which are interpreted by the remote computer, and view the resulting output on your computer. Although a few web browsers, such as Lynx, can operate via a shell account, they don't generally support the highly graphical, multimedia pages which web surfers have come to expect.

In contrast, a PPP account connects your computer directly to the Internet. While your computer is connected to the Internet, you can use it to surf the Web with your favorite browser. If your ISP allows, you can even run a web server, providing pages that can be viewed by others around the world.

You can compare the two types of Internet accounts - shell and PPP - with two kinds of postal service. Imagine that no mail carrier actually comes to your home to pick and deliver mail. Instead, every time you want to conduct postal business, you go to the post office. This resembles a shell account: The computer that connects you to the Internet is remote, and every time you want to do something on the Internet you must open a terminal, or telnet, session to that computer. PPP, on the other hand, is like home delivery: The Internet comes right to your doorstep, and your computer is literally placed on the Internet by the machine at your ISP that you connect to.

Under Microsoft Windows, you use `hyperterminal` to access a shell account and Dial-Up Networking to access a PPP account. Under Linux, you can choose from among several programs that let you access a shell account. The most commonly used programs are `minicom` and `seyon`. To access a PPP account under Linux, you use

the PPP daemon, `pppd`. The next section describes how to use `wvdial` to make the process of establishing a PPP connection simple.

11.2 Configuring Your Modem

Your modem is a peripheral device to Linux, like a CD-ROM drive, hard drive, or a mouse. Your modem will be connected to a serial port, which means that one of the serial devices in the */dev* directory actually points to your modem. You may notice that there's also a file, */dev/modem*, which serves as a placeholder for your modem. Initially this file doesn't point to anything. To simplify your operations, so that applications like `minicom` and `seyon` need only refer to */dev/modem*, you can create a symbolic link from the serial device connected to your modem to */dev/modem*.

To do so, login as `root` and issue the command:

```
ln -sf /dev/ttyS  
X /dev/modem
```

where *x* is replaced by the number corresponding to the serial device that points to your modem. [Table 11.1](#) shows how the proper command corresponds to the device by which your modem is known user DOS and Linux.

Table 11.1: Command to Create */dev/modem*

DOS Modem Device	Linux Modem Device	Command
COM1	ttyS0	<code>ln -sf /dev/ttyS0 /dev/modem</code>
COM2	ttyS1	<code>ln -sf /dev/ttyS1 /dev/modem</code>

DOS Modem Device	Linux Modem Device	Command
COM3	ttyS2	<code>ln -sf /dev/ttyS2 /dev/modem</code>
COM4	ttyS3	<code>ln -sf /dev/ttyS3 /dev/modem</code>

11.3 Using wvdial

ISPs use a variety of dialogs to make a PPP connection. Often the most difficult part of configuring your computer to make a PPP connection is specifying dialog options consistent with those required by your ISP, which may not be especially helpful in explaining the necessary options to you.

For example, most ISPs use one of three PPP login procedures: PAP (Password Authentication Protocol), CHAP (Challenge-Handshake Authentication Protocol), or ordinary user/password authentication. PAP is currently the most popular of the three procedures. If you ask the tech support person at your ISP what login procedure your ISP uses, the tech support person may have no idea what you're talking about, because most users run Microsoft Windows, and Windows Dial-Up Networking handles the login procedure transparently.

The `wvdial` program transparently establishes a PPP connection in much the same way as Windows Dial-Up Networking. It understands a variety of possible dialogs used by ISPs. In most cases, it will analyze data sent by your ISP and respond with the proper data in the format required by the ISP.

To use `wvdial`, be sure that you've established your nameserver configuration, as described in **Chapter 10**,

Setting Up a Linux-Based LAN. Then, issue the following commands

```
route del default  
wvdial &
```

The command generates quite a bit of output, which makes further use of this virtual terminal distracting. The simplest solution is to switch to another virtual terminal, by pressing **Alt-*n***, where *n* stands for the virtual terminal (1-7). Alternatively, you can direct the output of the command to a file, by typing this command in place of the one given earlier:

```
wvdial 2>/tmp/wvdial.messages &
```

Of course, you'll need to consult the file if something goes wrong with `wvdial`. Do so by using the `more` command:

```
more /tmp/wvdial.messages
```

Once your connection is up, you can browse the Web and access other Internet services, as described later in this chapter. For now, simply verify that your connection is working by issuing the command:

```
ping www.oreilly.com
```

The `ping` command should report that echo packets were successfully received from the server. If not, check your nameserver configuration.

When you want to log off your ISP, issue the following command:

```
killall wvdial
```

11.4 PPP Client

The Linux PPP client lets your Linux system connect to the Internet via a PPP server, much the same way Dial-Up Networking lets your Microsoft Windows system do so. Behind the scenes, `wvdial` calls on the services of the PPP client to connect to a PPP server. However, it's possible to use the Linux PPP client directly, without the aid of `wvdial`. Doing so gives you access to the many options provided by PPP, which can prove helpful if your networking needs are too sophisticated to be met by `wvdial`'s simple fire-and-forget mechanism. Even if you're content with `wvdial`, you'll benefit from understanding the PPP client. For example, you'll find it much easier to troubleshoot PPP problems if you understand the functions performed by the PPP client on behalf of `wvdial`.

11.4.1 Specifying PPP Options

The Linux PPP client requires several configuration files. Rather than build these files manually, you can use `pppconfig`, a program that leads you through a dialog and then creates the proper configuration files based on your responses. Login as `root`, and issue the command:

```
pppconfig
```

[Figure 11.1](#) shows `pppconfig`'s main menu, which the program displays whenever you start it. Notice that `pppconfig` lets you create a connection - the term by which it refers to a complete set of configuration parameters - but it does not let you edit an existing connection. To change a connection, you must edit the configuration files manually. It's usually simpler to forego editing a connection, creating a new connection instead.

Figure 11.1: The `pppconfig` main menu

 Figure 11.1

To create a new connection, use the arrow keys to highlight Next or Create, and then press **Enter**.

If `pppconfig` determines that the file `/etc/resolv.conf` does not exist or does not specify one or more nameservers (DNS servers), it will display the screen shown in [Figure 11.2](#), which asks you to specify nameservers. You should do so, therefore highlight Yes and press **Enter**.

Figure 11.2: Request to specify nameservers

 Figure 11.2

[Figure 11.3](#) shows the screen used to specify the primary nameserver. To specify the primary nameserver, type its IP number and press **Enter**. Your ISP should be able to tell you what IP number to specify.

Figure 11.3: Specifying the primary nameserver

 Figure 11.3

You can also specify the IP number of a secondary nameserver, as shown in [Figure 11.4](#). Simply type the IP number and press **Enter**. Your ISP should be able to provide you with the IP number of the secondary nameserver; if you don't know the IP number, leave it blank.

Figure 11.4: Specifying the secondary nameserver

 Figure 11.4

[Figure 11.5](#) shows the next screen displayed by `pppconfig`, which asks for a name by which to refer to this connection. If your `/etc/resolv.conf` file already specified one or more nameservers, `pppconfig` displays this screen after its initial screen. Choose a connection name you'll be able to recall, backspace over the default

connection name, type your connection name, and press **Enter**. You may find it convenient to use the domain name of your ISP as the connection name.

`pppconfig` equivocates by sometimes referring to a *connection*, sometimes to an *ISP* (Internet Service Provider), and sometimes to a *provider*. The term *connection* is more accurate in this context, because you may have several ways of connecting to a particular ISP - for example, the ISP may have several dial-up numbers. Each way of connecting to an ISP can be configured as a distinct `pppconfig` connection.

Figure 11.5: Specifying the connection (provider) name

 Figure 11.5

Next, `pppconfig` asks what method your ISP uses to authenticate your access to its network, as shown in [Figure 11.6](#). PAP (Password Authentication Protocol) is currently the most popular method; however, some ISPs continue to use other methods. The technical support staff of many ISPs cannot tell what authentication method their system uses. So, you may have to guess; if so, initially guess PAP. If you find you cannot establish a connection, you can try the alternative values. Select the desired authentication method and press **Enter**.

Figure 11.6: Specifying the authentication method

 Figure 11.6

Next, you must specify your login password, as shown in [Figure 11.7](#). Type the password and press **Enter**.

Figure 11.7: Specifying the password

 Figure 11.7

Next, `pppconfig` asks you to specify some basic connection properties. Press **Enter** to exit the Configuration screen and the Manager User Configuration screen, shown in [Figure 11.8](#), appears. Type the login username assigned to you by your ISP and press **Enter**. Your login username may differ from your actual username, so be sure to use your login username.

Figure 11.8: Specifying the username

 Figure 11.8

Next, you must specify the device that corresponds to your modem, as shown in [Figure 11.9](#). Type the name of the device file, paying careful attention to capitalization, and press **Enter**.

Figure 11.9: Specifying the modem device

 Figure 11.9


The program next asks whether you want it to establish a default route when it makes a connection, as shown in [Figure 11.10](#). You almost certainly do, so accept the default choice - Enable Default Route - by using the arrow keys to highlight Ok and pressing **Enter**.

Figure 11.10: Enabling a default route

 Figure 11.10

[Figure 11.11](#) shows the next `pppconfig` screen, which requests information on your computer's network address (IP number). Most ISPs assign your computer a different network address each time you establish a connection. Some ISPs give you the option of having a static network address, meaning that your computer's network address is always the same.

Figure 11.11: Specifying your computer's network address

 Figure 11.11

Unless your ISP has assigned your computer a static network address, simply press **Enter**. If your computer has a single static network address, type the address followed by a colon and press **Enter**. If it has separate local and remote network addresses, type the local address, a colon, and the remote address; then press **Enter**.

Next, you must specify your modem's speed, as shown in [Figure 11.12](#). This refers to the speed of the connection between your modem and your system, not to the speed of the connection between your modem and your ISP, which is normally a smaller value. Unless you have a particular reason for doing otherwise, you should accept the default speed of 115200 by pressing **Enter**.

Figure 11.12: Specifying the modem speed

 Figure 11.12

Next, as shown in [Figure 11.13](#), `pppconfig` lets you specify a modem initialization string, which it will send to the modem before establishing a connection. Simply press **Enter** to accept the default initialization string, which works correctly for most modems and situations.

Figure 11.13: Specifying a modem initialization string

 Figure 11.13


Next, as shown in [Figure 11.14](#), you must specify the number your modem should dial to reach your ISP. Type the number without spaces or punctuation. With the number, include any necessary prefix or suffix digits just as you would dial them yourself - for example, the *1* and area code needed to access a long distance number.

Figure 11.14: Specifying the phone number

 Figure 11.14

Next, as shown in [Figure 11.15](#), you must specify whether your phone line accepts tone dialing or pulse dialing. Most U.S. phone lines accept tone dialing; unless you're certain your line does not, specify tone dialing, highlight Ok, and press **Enter**.

Figure 11.15: Specifying tone or pulse dialing

 Figure 11.15

Next, `pppconfig` gives you the opportunity to review and change your entries. Press **Enter** to exit the Properties screen. [Figure 11.16](#) shows the screen that appears. To view or change an entry, highlight the entry and press **Enter**. When you're satisfied that your entries are correct, highlight Finished and press **Enter**.

Figure 11.16: Reviewing and changing entries

 Figure 11.16

`pppconfig` writes the PPP configuration files according to your specifications. Press **Enter** to exit the Finished screen. Unless you want to specify another connection, highlight Quit and press **Enter** to exit the main menu.

11.4.2 Establishing a PPP Connection

Finally, you're ready to establish a PPP connection. While logged in as `root`, issue the following command:

```
pon  
connection
```

Where `connection` is the connection name you earlier specified in `pppconfig`.

You can monitor the progress of the operation by issuing the following command in a separate virtual console:

```
tail -f /var/log/messages
```

This command continually displays messages as they're posted to the system log file. Because PPP writes log entries that describe its progress, the command lets you see what PPP is doing. When you're done viewing log entries, type **Ctrl-C** to exit the `tail` command.

You can see even more detail by using the command:

```
tail -f /var/log/ppp.log
```

You can verify that the PPP connection has been established by issuing the following command:

```
ifconfig
```

If the output of the command includes a description of a PPP connection, the connection was successfully established. To verify that the connection is operational, issue the `ping` command:

```
ping www.oreilly.com
```

To terminate the connection, type the command:

```
poff
```

If your connection failed, you should check the following before contacting your ISP or other Linux users for support:

- Did you correctly specify your ISP's phone number?
- Did you correctly specify your userid and password?
- Did you correctly specify your modem's device file and line speed?
- Is your modem properly connected to your telephone line?

- Is your telephone line operational?


When you want to terminate the PPP connection, you can click on the Disconnect button of the dialog box you used to establish the connection.

11.5 Web Browser

Once you've established a PPP connection to the Internet, whether via `wvdial` or `pppd`, you can surf the Web using a Linux browser. In addition to browsers written specifically for Unix and Linux, you can use the popular Netscape Navigator web browser, which includes a mail client, newsgroup client, HTML editor, and other features.

As you can see in [Figure 11.17](#), the Linux version of Navigator is very similar to the Microsoft Windows version. So, if you've used the Microsoft Windows version of Navigator, you'll find configuring and using Navigator to be quite straightforward. To configure Navigator, click on Edit → Preferences. Then specify your identity and that of your mail and newsgroup servers, along with any other special preferences you desire.

Figure 11.17: Netscape Navigator for Linux

 Figure 11.17

11.6 gFTP FTP client

You can use your web browser to download files from an FTP server, but to upload files you need an FTP client. The `gFTP` client, included on the Linux CD-ROM, is an excellent choice, because its user interface resembles that of popular Microsoft Windows FTP clients, such as WS-FTP.


Figure 11.18 shows the `gFTP` client. To connect to a remote system, click on Remote → Connect, identify the system's hostname, specify any necessary userid or password, and click on Connect. To upload a file, click on the name of the file in the local list box at the left of the window and then click on →. To download a file, click on the name of the file in the list box at the right of the window and then click on  <-. When you've transferred all your files, click on Remote → Disconnect.

Figure 11.18: The gFTP FTP client

 Figure 11.18

11.7 Using minicom and seyon

Some ISPs provide a choice of a PPP account or a shell account. The two most popular Linux programs for accessing a shell account are `minicom`, a graphical program that runs in a virtual console, and `seyon`, which runs under X. Although `seyon` has the more modern user interface, most Linux users prefer `minicom`, which supports more options than does `seyon`.

11.7.1 Using seyon

Once you've configured your modem, you're ready to run `seyon`. To do so, start X and issue the command:

```
seyon
```

[Figure 11.19](#) shows the main `seyon` window. The top row contains several labels that indicate the status of your modem:

DTR

Indicates that your computer is ready to send and receive data

DSR

Indicates that the modem is ready to send and receive data

RTS

Indicates that your computer has requested to send data

CTS

Indicates that the modem is ready to receive data

RNG


Indicates that the modem has detected a ring signal on the telephone line

Figure 11.19: The main *seyon* window

 Figure 11.19

The second row contains several programmable buttons, which have no associated default action. The remaining buttons, in the third, fourth, and fifth rows, let you operate *seyon*. For example, to make a connection, you click on the Dial button, which pops up *seyon*'s Dialing Directory dialog box, shown in [Figure 11.20](#).

Figure 11.20: The *seyon* Dialing Directory

 Figure 11.20

However, the Dialing Directory dialog box will not appear until you create a file named *phonelist* in the *.seyon* subdirectory of your home directory. To do so, issue the following commands:

```
cd
mkdir .seyon
> .seyon/phonelist
```

The third command uses a clever trick to create an empty file: it redirects the output of a null command to the file. Now, when you click on Dial, the Dialing Directory dialog box appears, although no entries will be shown. To create an entry, click on Edit and an edit window appears. Type the name of the entry and the phone number, separated by a space. If you like, you can specify other options, such as the desired connection speed, the number of bits per character, the data parity, and the number of stop bits associated with a character. However, usually such options are not required. Clicking on Help on *seyon*'s main menu will pop up a window that describes *seyon* and its operation, including the contents of the *phonelist* file.

To initiate a connection, simply highlight the entry in the Dialing Directory dialog box and click on Go. When the connection has been established, the Dialing Directory dialog box will disappear. You can click Hangup to abruptly terminate a connection or Exit to exit `seyon`.

11.7.2 Using `minicom`

Despite a user interface less modern than that of `seyon`, `minicom` is the more popular Linux communications program. Before launching `minicom`, you must configure it. To do so, login in as `root` and issue the command:

```
minicom -s
```

[Figure 11.21](#) shows the Configuration dialog box that `minicom` presents. Like the Linux install program, `minicom` does not support use of the mouse. Instead, you use tab and cursor keys to navigate the screen. For example, you can use the up and down arrow keys to highlight various items on the configuration menu. To select an item, highlight it and press **Enter**.

Figure 11.21: The `minicom` configuration dialog box

 Figure 11.21


Generally, the default configuration options are acceptable. You can simply highlight Save Setup As Dfl, press **Enter**, then highlight Exit From Minicom, and press **Enter**. However, the next several subsections briefly describe the available options just in case you may need to change them.

11.7.2.1 Configuring filenames and paths

The `minicom` dialog box for configuring filenames and paths is shown in [Figure 11.22](#). Using it, you can configure the default directories used by `minicom` for downloads, uploads, and scripts. You can also

configure the name of the program used to process `minicom` scripts and the path for `kermit` (a program used to perform file transfers) though it's unlikely you'll want to do so.

Figure 11.22: Configuring filenames and paths

 Figure 11.22

To change an option, type the letter that appears to the left of the option, type the desired value of the option, and press **Enter**. No changes are stored until you select a Save item from the main menu.

11.7.2.2 Configuring file transfer protocols

The `minicom` dialog box for configuring file transfer protocols is shown in [Figure 11.23](#). For each supported protocol, you can specify the path and command line arguments used to perform uploads and downloads. You can also specify how ASCII uploads are performed and set several transfer options.

Figure 11.23: Configuring file transfer protocols

 Figure 11.23

The columns have the following meanings:

- Name specifies whether the program requires a filename as an argument
- U/D specifies whether a given row specifies an upload protocol (U) or download protocol (D)
- FullScr specifies whether the program runs in its own window
- IO-Red. specifies whether the program obtains its input from the standard input stream and directs its output to the standard output stream

- Multi specifies whether multiple files can be transferred by a single command

As mentioned, you probably won't need to change any file transfer options. However, you can change an option by typing the letter that appears to its left, typing the desired value of the option, and pressing **Enter**. No changes are stored until you select a Save item from the main menu.

11.7.2.3 Configuring the serial port

The `minicom` dialog box for configuring the serial port is shown in [Figure 11.24](#). If you have a high-speed modem, you may obtain faster data transfers by increasing the data rate from the default 38400 bps to 57600 bps or 115200 bps. You can change this and other options in the same way you change filename and path options and file transfer protocol options.

Figure 11.24: Configuring the serial port

 Figure 11.24

11.7.2.4 Configuring modem and dialing options

The `minicom` dialog box for configuring modem and dialing options is shown in [Figure 11.25](#). If your modem uses standard (Hayes-compatible) commands, you'll probably find the default options satisfactory. However, the default options will probably not take advantage of special capabilities afforded by your modem. You can consult your modem documentation and revise the command strings to incorporate the codes that activate your modem's special features.


Figure 11.25: Configuring modem and dialing options

 Figure 11.25

11.7.2.5 Configuring screen and keyboard options

The `minicom` dialog box for configuring screen and keyboard options is shown in [Figure 11.26](#). Many users prefer screen colors other than those provided by default. You can separately specify the foreground and background colors of the menu, terminal window, and status line.

Figure 11.26: Configuring screen and keyboard options

 Figure 11.26

11.7.2.6 Saving your changes

After you've made changes to `minicom`'s options, use the Save Setup As Dfl item on the main menu to save your new configuration as the default configuration. To do so, highlight the Save Setup As Dfl item, press **Enter**, then highlight Exit From Minicom, and press **Enter**.

11.7.2.7 Running minicom

Once you've configured `minicom`, you're ready to launch it. To do so, issue the command:

```
minicom -c on
```

The arguments specify that `minicom` should present a color display; without them, its output is monochrome. [Figure 11.27](#) shows `minicom`'s terminal window and status line.


Figure 11.27: The minicom main window

 Figure 11.27

To operate `minicom`, you use single letter commands that you call up by typing **Ctrl-A** and then the command. For example, typing **Ctrl-A**


followed by **Z** produces the command summary shown in [Figure 11.28](#).

Figure 11.28: The minicom Command Summary

 Figure 11.28

To access `minicom`'s Dialing Directory dialog box, type **Ctrl-A** followed by **D**. [Figure 11.29](#) shows this dialog box. The menu at the bottom of the dialog box lets you dial the selected entry, find an entry that contains a specified text string, add a new entry, edit an existing entry, or remove an entry. You can also manually dial a number that's not in the directory.

Figure 11.29: The minicom Dialing Directory

 Figure 11.29

Once you've mastered `minicom`'s basic functions, you might enjoy learning how to write scripts. Using scripts, you can (for example) automate logging into your shell account, eliminating the need for you to recall your userid and password. Consult the files in `/usr/doc/minicom/doc` and `/usr/doc/minicom/demos` to learn how `minicom`'s scripting language works and how to write your own scripts.

11.8 Making a PPP Connection Manually

Sometimes you may have difficulty making a PPP connection by using `wvdial`. In such a case, it's helpful to know how to make a PPP connection manually. Once you make a connection, you can use email and Usenet newsgroups to seek the help you need to resolve your problem in using `wvdial`.

Before making the PPP connection, you must have:

- Established the name server configuration, as described in **Chapter 10**.
- Established the PAP configuration, done for you by `pppconfig` as described earlier in this chapter.

If you're unsure whether you've performed these operations, do them again, just to be safe.

To make a PPP connection manually, log in as `root`, launch `minicom` and dial your ISP. If your ISP prompts for a `userid` or `password`, reply appropriately. Most ISPs now use automatic authentication, so it isn't usually necessary to enter your user information. Once authentication is complete, nonsense characters should begin to appear on `minicom`'s screen. Type **Ctrl-A** then **J** to suspend `minicom` and launch a shell. At the shell prompt, issue the commands:

```
route del default
pppd -detach defaultroute /dev/modem 38400 &
```

In a short while, the PPP connection should be made. To verify that the connection is working, use a web browser to contact a remote server. If this doesn't work, check your configuration carefully.

To shut down the PPP connection, issue the command:

```
poff
```


After a short pause, the modem will hang up. To resume `minicom`, type the command:

`fg`

You can then exit `minicom` normally.



Learning Debian GNU/Linux

By Bill McCarty

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

12. Setting Up a Linux-Based WAN

Contents:

[An FTP Server](#)

[Installing and Configuring a Web Server](#)

[Using a Mail Server](#)

[Configuring a Secure Shell Server](#)

[Configuring a Dial-In Shell Server](#)

In the last chapter, you learned how to connect your Linux system to a local-area network or, via an Internet Service Provider, to the Internet. By doing so, you were able to access a plethora of services provided by others, including file transfers via FTP, web pages, email, and telnet. In this chapter you'll learn how to set up and use several Linux wide-area network servers, including an FTP server, a web server (Apache), an email (SMTP/POP) server, and a dial-in shell server. These applications let you and others access data on your Linux system from anywhere in the world via the Internet. These applications will be most useful if your system is connected to the Internet 24 hours a day, 7 days a week. But, even if your connection is intermittent, you and others can access the services these applications provide whenever the connection is active.

12.1 An FTP Server

An FTP server lets you transfer files from one system to another, via a network. When two computers are connected to the Internet, you can use FTP to transfer files from one to the other even though the computers are not directly connected.

An FTP server attempts to authenticate users that request to use it. You can configure your FTP server to accept requests only from users who have an account on the system running the FTP server. Alternatively, you can configure the FTP server to accept requests from anyone, via a facility known as *anonymous FTP*. It's fairly simple to install and configure an anonymous FTP server; however, hackers regularly exploit vulnerabilities in anonymous FTP servers, breaking into systems and causing manifold mischief. Because it's difficult to protect a system running anonymous FTP from attack, this section does not describe the process for installing and configuring anonymous FTP.

Selecting the Basic profile during Debian GNU/Linux installation causes installation of a standard FTP server.

12.1.1 Testing the FTP Server

To test your FTP server, start an FTP client by issuing the following command:

```
ftp localhost
```

The FTP server should prompt you for a login userid and password. If you correctly supply them, you should see the FTP prompt that lets you know the FTP server is ready to execute FTP subsystem commands. Type `quit` and press **Enter** to exit the FTP client. Or, if you'd like to transfer some files, you can use the FTP subsystem commands described in [Table 12.1](#).

Table 12.1: Important FTP Subsystem Commands

Command	Function
<code>! command</code>	Invokes a shell on the local system. You can use this command, for example, to obtain a listing of the current directory on the local system by issuing the command <code>!ls</code> , for a Unix system, or <code>!dir</code> , for a Microsoft system.

Command	Function
<code>ascii</code>	Specifies that files will be transferred in ASCII mode.
<code>binary</code>	Specifies that files will be transferred in binary mode, which performs no translation.
<code>cd directory</code>	Changes to the specified directory of the remote system.
<code>delete file</code>	Deletes the specified file from the remote system.
<code>dir</code>	Displays the contents of the current directory of the remote system.
<code>get file</code>	Retrieves the specified file from the remote system.
<code>help</code>	Displays command help information.
<code>lcd directory</code>	Changes to the specified directory of the local system.
<code>mkdir directory</code>	Creates the specified directory on the remote system.
<code>put file</code>	Stores the specified local file on the remote system.
<code>pwd</code>	Displays the current working directory on the remote system.
<code>quit</code>	Exits the FTP subsystem.
<code>rmdir directory</code>	Removes the specified directory from the remote system.

If your FTP server fails to respond properly, check the line you added to the *inetd.conf* file. If you're unable to find an error, reboot your system. If that fails to solve the problem, post a message to the *comp.os.linux.setup* newsgroup.

Once your FTP server is working, try contacting it from a remote system. If you have a Microsoft Windows system, you can contact your server by using the built-in FTP client that works similarly to the Linux FTP client, interpreting the same FTP subsystem commands. Open an MS-DOS Prompt window and type the command:

```
ftp  
server
```

where *server* specifies the hostname or IP address of your Linux server. Generally, once the FTP subsystem prompt is available, you should immediately issue the *binary* command. This command specifies that files will be transferred verbatim; without it, executable files, documents, and other files that contain binary data will be scrambled when transferred.

Most Windows users prefer to use a graphical FTP client. Many such clients, including WS-FTP, are freely available and make FTP access easy for even novice Windows users. FTP provides a very fast and reliable way for a Linux server to share files with Windows clients, without the need to install and configure Samba.

12.2 Installing and Configuring a Web Server

Installing and configuring a web server is not much more difficult than installing an FTP server. Once your web server is up and running, other Internet users can view documents you publish on your Linux system.

12.2.1 Configuration

Configuring a web server can be as easy or as difficult as you choose. Like other web servers, Apache provides seemingly countless options. Fortunately, Debian GNU/Linux automatically configures Apache when you install it. However, the configuration options selected by the install scripts may not suit your needs. In that case, you can modify the configuration files.

Apache's configuration files reside in the directory */etc/apache*. For historical reasons that no longer apply, Apache has three configuration files:

`access.conf`

Specifies what hosts and users are allowed access to what documents and services

`httpd.conf`

Specifies options that govern the operation of the `httpd` daemon

`srm.conf`

Specifies how your server's documents are organized and formatted

Currently, you can place Apache configuration commands in any of these files. However, each of these files must exist, even if it is empty; otherwise, the `httpd` daemon will refuse to run. As distributed, the files contain a default configuration. Before starting the web server, you should revise the `ServerName` option of the *httpd.conf* file. The three following subsections describe other options that you may wish to specify. You can scan them to see what options are available and specify options that interest you. A more complete description of the options is available in Apache's online documentation. Also, the Apache web site (<http://www.apache.org/>) provides a tutorial on Apache configuration. To change an option, simply open the related file by using your favorite text editor, change the file as you wish, and save the file.

The subsections assume some familiarity with HTML and web servers. If you find that some options are obscure, don't fret; your web server will serve ordinary HTML

pages even if you set no options other than `ServerName`.

12.2.1.1 The `access.conf` file

The `access.conf` file specifies a default set of permissions that govern access to documents and services. It then specifies sets of permissions that override the default permissions for particular documents and services. The usual practice is to specify a quite restrictive set of default permissions, relaxing these permissions to provide access to particular documents and services.

The file contains a mixture of comments (lines beginning with `#`) and directives. Comments are ignored by the server.

The default permissions are specified as follows:

```
<Directory />
Options None
AllowOverride None
</Directory>
```

The paired tags `<Directory>` and `</Directory>` enclose a list of options that pertain to the / directory, the directory specified in the `<Directory>` tag. The options are:

- `Options None`, which specifies that no special server features are enabled for the specified directory or its subdirectories.
- `AllowOverride None`, which specifies that access specifications cannot be overridden by an `.htaccess` file.

[Table 12.2](#) describes special server features that are available.

Table 12.2: Special Server Features

Option	Description
<code>ExecCGI</code>	Execution of CGI scripts is permitted in this directory.
<code>FollowSymLinks</code>	The server will follow symbolic links in this directory.
<code>Includes</code>	Server-side includes are permitted.

Option	Description
<code>IncludesNOEXEC</code>	Server-side includes, except <code>#exec</code> and <code>#include</code> , are permitted in this directory.
<code>Indexes</code>	If the directory contains no index file (for example, <i>index.html</i>), the server will prepare a formatted index.
<code>MultiViews</code>	Content-negotiated MultiViews are permitted in this directory. MultiViews permit, for example, a client browser to select a document in a particular language from a set of documents.
<code>SymLinksIfOwnerMatch</code>	The server will follow symbolic links for which the target file or directory has the same owner as the link.

Unless the specification for a directory specifies `AllowOverride None`, you can override the specified options by placing an *.htaccess* file in the directory or one of its subdirectories. The *.htaccess* file can contain specifications of the same sort as the *access.conf* file; the server applies the specifications in the *.htaccess* file in preference to those specified in the *access.conf* file.

After the restrictive default specifications come some more relaxed specifications:

```
<Directory /home/httpd/html>
Options Indexes Includes FollowSymLinks
AllowOverride None
order allow,deny
allow from all
</Directory>
```

These specifications apply to the directory */home/httpd/html* and its subdirectories. Here, the `Indexes`, `Includes`, and `FollowSymLinks` options are specified. As for the root directory, use of *.htaccess* files is forbidden, via `AllowOverride None`. Here, unlike the specification for the root directory, the hosts allowed to access documents are services are specified. The `order allow,deny` directive specifies that any `deny` directives will be applied after any `allow` directives, and will therefore take

precedence. No `deny` directives appear in this specification; the `allow from all` directive permits any host to access documents and services within the `/home/httpd/html` directory and its subdirectories.

Another specification allows execution of CGI scripts within the `/home/httpd/cgi-bin` directory and its subdirectories:

```
<Directory /home/httpd/cgi-bin>
AllowOverride None
Options ExecCGI
</Directory>
```

The final specification in the default configuration lets the local host access HTML documents within the `/usr/doc` directory and its subdirectories:

```
<Directory /usr/doc>
order deny,allow
deny from all
allow from localhost
Options Indexes FollowSymLinks
</Directory>
```

12.2.1.2 The `srm.conf` File

The `srm.conf` file specifies the organization and format of documents provided by your web server. As was the case with the `access.conf` file, you don't need to make any changes to the `srm.conf` file, though you may wish to do so.

The `DocumentRoot` directive specifies the directory that contains your HTML files. When a web client accesses the root directory, the server actually fetches files from the directory specified as `DocumentRoot`:

```
DocumentRoot /home/httpd/html
```

The `UserDir` directive specifies the name of the subdirectory that the server appends to a user's home directory when a client makes a `~user` request:

```
UserDir public_html
```

This directive specifies that a reference to `~user` will be translated to a reference to `/home/user/public_html`.

The `DirectoryIndex` directive specifies the name of the file (or names of the files) used as directory indexes:

```
DirectoryIndex index.html index.shtml index.cgi
```

The `FancyIndexing` directive specifies whether icons are used to produce fancy directory indexes:

```
FancyIndexing on
```

The `AddIcon` and `AddIconByType` directives associate icons with files of given types:

```
AddIconByType (TXT,/icons/text.gif) text/*  
AddIcon /icons/binary.gif .bin .exe
```

The default configuration includes many such directives. If you wish to add support for a new type of file, you may want to add a directive associating an icon with the new file type.

The `DefaultIcon` directive specifies the icon used for file types not explicitly associated with an icon:

```
DefaultIcon /icons/unknown.gif
```

The `ReadmeName` directive specifies the name of a file used by the server to produce readme entries:

```
ReadmeName README
```

The server will first look for the file `README.html` and then for the file `README`.

The similar `HeaderName` directive specifies the name of a file that the server will prepend to a generated index:

```
HeaderName HEADER
```

The `IndexIgnore` directive specifies a set of file names that should not be included in a generated index. These are often specified by using wildcard characters:

```
IndexIgnore .??* *~ *# HEADER* README* RCS
```

The `AccessFileName` directive specifies the name of the file that, if present, overrides access control specifications for a directory:

```
AccessFileName .htaccess
```

The `TypesConfig` directive identifies the *mime.types* file, which describes multimedia files:

```
TypesConfig /etc/mime.types
```

The `DefaultType` directive specifies the default MIME type for documents:

```
DefaultType text/plain
```

The `AddEncoding` directive instructs compatible browsers to uncompress information as it's downloaded:

```
AddEncoding x-compress Z
AddEncoding x-gzip gz
```

The `Redirect` directive lets you provide a forwarding address for documents that have moved. The default configuration includes no `Redirects`, which have this simple form:

```
Redirect old-URL new-URL
```

The `Alias` directive lets you refer to a directory by using an alias. For example, the following directive provides a more convenient way of referring to the </home/httpd/icons> directory:

```
Alias /icons/ /home/httpd/icons/
```

The `ScriptAlias` directive lets users refer to the CGI directory as simply </cgi-bin/>:

```
ScriptAlias /cgi-bin/ /home/httpd/cgi-bin/
```

The following sets of specifications work around problems with several browsers. First, two directives that disable `keepalive` for browsers that do not support it:

```
BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0
```

The final directives force use of HTTP version 1.0 responses for several browsers that do not support HTTP 1.1:

```
BrowserMatch "RealPlayer 4\.0" force-response-1.0
BrowserMatch "Java/1\.0" force-response-1.0
BrowserMatch "JDK/1\.0" force-response-1.0
```

12.2.1.3 The `httpd.conf` File

The *httpd.conf* file specifies options related to the `httpd` daemon. You should specify the `ServerName` option before starting your web server.

The `ServerType` directive specifies whether the web server is started via `inetd` or standalone:

```
ServerType standalone
```

The `port` directive specifies the port on which the web server listens for client requests:

Port 80

The `HostnameLookups` directive specifies whether clients are logged by IP address (off) or hostname (on):

```
HostnameLookups off
```

The `User` and `Group` directives specify the userid and group under which the `httpd` daemon runs. The daemon initially runs as `root` and then switches to the specified userid and group. The default configuration specifies the userid as `nobody`, a standard Unix userid that has very limited permissions. A user or process running as `nobody` can access files only in ways permitted to all users; generally, this means the user or process cannot modify files. The group `nobody` has similarly constrained privileges:

```
User nobody
Group nobody
```

The `ServerAdmin` directive specifies the email address of the server administrator:

```
ServerAdmin root@localhost
```

The `ServerRoot` directive specifies the directory that contains the configuration, error, and log files:

```
ServerRoot /etc/httpd
```

The `BindAddress` directive provides for virtual hosts. It specifies the IP address to which the server should listen. It is normally disabled by a comment token:

```
#BindAddress *
```

The `ErrorLog` directive specifies the location of the error log file:

```
ErrorLog logs/error_log
```

The `LogLevel` directive specifies the verbosity of the server log. Possible values include `debug`, `info`, `notice`, `warn`, `error`, `crit`, `alert`, and `emerg`:

```
LogLevel warn
```

The `LogFormat` directive specifies format names that can be used with the `CustomLog` directive:

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
CustomLog logs/access_log common
```

For more information about `LogFormat` and `CustomLog`, see the Apache documentation.

The `PidFile` directive specifies the file in which the server should log its process id:

```
PidFile /var/run/httpd.pid
```

The `ScoreBoardFile` directive specifies the name of the file used to store internal server process data:

```
ScoreBoardFile /var/run/httpd.scoreboard
```

The `LockFile` directive specifies the file used to providing locking. You'll need to change this option only if you NFS mount the directory used by the server for logging:

```
#LockFile /var/lock/httpd.lock
```

The `ServerName` directive specifies the hostname of the system on which your server runs. Depending on your network configuration, you may not need to specify this directive:

```
ServerName host.domain.com
```

The `UseCanonicalName` directive specifies whether the server will return a canonical URL formed from the `ServerName` and `Port` directives (on) or the hostname and port supplied by the client (off):

```
UseCanonicalName on
```

The `CacheNegotiatedDocs` directive instructs browsers not to cache documents; it is usually disabled by prefixing it with a comment token (#):

```
#CacheNegotiatedDocs
```

The `Timeout` directive specifies the maximum number of seconds the server will wait for certain responses, such as the next packet in a sequence of TCP packets:

```
Timeout 300
```

The `KeepAlive` directive specifies that connections are persistent; that is, that a client can make multiple requests per connection:

```
KeepAlive On
```

The `MaxKeepAliveRequests` directive specifies the maximum number of requests permitted during a persistent connection:

```
MaxKeepAliveRequests 100
```

The value 0 denotes an unlimited number of requests.

The `KeepAliveTimeout` directive specifies the maximum number of seconds during which the server will wait for the next request:

```
KeepAliveTimeout 15
```

The `MinSpareServers` and `MaxSpareServers` directives respectively fix the minimum and maximum number of spare server processes the server will create. Having an available server expedites handling of an incoming request:

```
MinSpareServers 8  
MaxSpareServers 20
```

The `StartServers` directive specifies the initial number of server processes:

```
StartServers 10
```

The `MaxClients` directive specifies the maximum number of simultaneous server processes. When this number is reached, requests from additional clients are locked out:

```
MaxClients 150
```

The default values of these options are higher than necessary for most workstation users. Your system will perform more efficiently (and perhaps more reliably) if you change the directives to specify the following values:

```
MinSpareServers 3  
MaxSpareServers 6  
StartServers 3  
MaxClients 20
```

The `MaxRequestsPerChild` directive specifies the number of requests a child process can handle before expiring. This ensures that processes are periodically recreated, minimizing problems due to software errors such as memory leaks:

```
MaxRequestsPerChild 100
```

The `ProxyRequests` directive specifies whether the proxy server is enabled; it is normally disabled:

```
#ProxyRequests On
```

When the proxy server is active, the following directives specify various caching options. They are normally disabled:

```
#CacheRoot /var/cache/httpd  
#CacheSize 5  
#CacheGcInterval 4
```

```
#CacheMaxExpire 24
#CacheLastModifiedFactor 0.1
#CacheDefaultExpire 1
#NoCache a_domain.com another_domain.edu joes.garage_sale.com
```

The `Listen` directive lets you bind Apache to a specific IP address or port, in addition to the default IP address and port. It is generally disabled:

```
#Listen 3000
#Listen 12.34.56.78:80
```

The `<VirtualHost>` and `</VirtualHost>` tags enclose a series of options that establish a virtual host, useful if your system has multiple IP addresses. The options can include any of the options described in this subsection. The tags and options are normally disabled:

```
#<VirtualHost host.some_domain.com>
#ServerAdmin webmaster@host.some_domain.com
#DocumentRoot /www/docs/host.some_domain.com
#ServerName host.some_domain.com
#ErrorLog logs/host.some_domain.com-error_log
#TransferLog logs/host.some_domain.com-access_log
#</VirtualHost>
```

12.2.2 Startup and Use

The install script will automatically start your web server as soon as you install it. If you installed the lynx browser, you can use it to test your web server. Issue the command:

```
lynx http://localhost
```


You should see a screen that resembles [Figure 12.1](#).

Figure 12.1: The Apache start page viewed by lynx

 Figure 12.1

If you prefer, you can view the start page by using Netscape Navigator, which supports the graphics embedded in the page. The result should resemble [Figure 12.2](#).

Figure 12.2: The Apache start page viewed by Netscape Navigator

 Figure 12.2

Once you can access your web server locally, try accessing it from a remote computer. This should be as simple as forming a URL that includes the fully

qualified hostname of your system (that is, the host and domain names); for example, <http://mysystem.mydomain>.

12.3 Using a Mail Server

The installation procedure installed `exim`, a Linux mail transfer agent. Therefore, you should be able to send email to users on other systems, and to receive email sent by them to you. Simply reconfigure your favorite mail client to specify your Linux system as your mail server. For example, if you're using Netscape Navigator, select Edit → Preferences → Mail Servers. Make a note of the existing mail server settings, just in case something goes wrong. Then, change the incoming and outgoing mail server options to specify your Linux machine.

If you want to continue receiving mail on another mail server, but want to read it from your Linux system, you might consider using `fetchmail`, a package that can retrieve email messages from the server (by using POP, IMAP, or just about any other remote mail protocol) and forward them to your Linux system.

12.4 Configuring a Secure Shell Server

A secure shell server lets you connect to a system from another system, via TCP/IP, and obtain a shell prompt, from which you can issue commands and view output. You may not be familiar with secure shell servers, but familiar with Telnet, which works similarly. A secure shell server differs from a Telnet server in that the conversation between a secure shell server and its clients is sent in encrypted form so that hackers cannot easily discover private information, including userids and passwords.

The secure shell client and server cannot be distributed on a CD, because they utilize advanced encryption techniques; U.S. law currently forbids general distribution of such software. The installation script will automatically start the `ssh` daemon.

12.4.1 Using a Secure Shell Client

To verify that the server is properly running, you can access it via a client on the local system. Issue the following command:

```
ssh localhost
```

The client will attempt to log you onto the local system by using your current userid, and should prompt you for your password. If you supply the correct password, you should see a shell prompt, indicating that the client and server are functioning correctly. Type `exit` and press **Enter** to exit the secure shell provided by the client.

12.4.2 Using a Secure Server from a Remote System

To log onto your Linux system from a remote system via the secure shell server, you must install a secure shell client on the remote system. A suitable client for Microsoft Windows 9x systems is `ttssh`, available from Robert O'Callahan's web site, <http://www.zip.com.au/~roca/ttssh.html>. Simply download and install `ttssh` on your Windows system, specify the hostname or IP address of your Linux system and your userid and password, and `ttssh` will log you onto your Linux system.

12.5 Configuring a Dial-In Shell Server

A dial-in server lets you connect to a system via a modem and phone line, and obtain a shell prompt, from which you can issue commands and view output. Using a dial-in server is a great deal like using Telnet or a secure shell server, except that your connection to the server is via a phone line. The `mgetty` package provides a simple-to-use dial-in server.

12.5.1 Configuration

To configure `mgetty` to answer incoming modem calls, use your favorite text editor to find the following lines in the file `/etc/inittab`:

```
# Example how to put a getty on a modem line
#
#T3:23:respawn:/sbin/mgetty -x0 -s 57600 ttyS3
```

Then, modify the last line by deleting the hash mark (`#`) and changing `ttyS3` to `modem`:

```
T3:23:respawn:/sbin/mgetty -x0 -s 57600 modem
```

Notice that the revised line invokes `mgetty`, not `getty`. The final argument specified on the added line assumes that your modem is configured as `/dev/modem`. If your modem is configured as a different device, you must adjust the added line accordingly. The **Chapter 11**,

Getting Connected to the Internet described the use of `modemtool` to configure your modem; if you followed its instructions, `/dev/modem` refers to your modem.

To cause your configuration change to take effect, you must reboot your system. Alternatively, you can cause the `init` process to re-examine the `inittab` file. To do so, issue the following command:

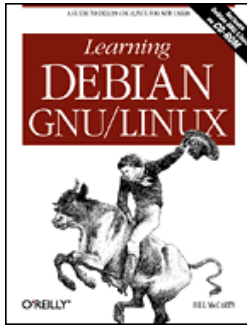
```
/sbin/telinit q
```

The `mgetty` process should begin monitoring your modem, awaiting an incoming call.

12.5.2 Using the Dial-In Server

To use the dial-in server, launch a program such as Microsoft Windows' Hyperterminal and place a call to the phone line to which your Linux system's modem is connected. Your Linux system should answer the call and provide you with a login prompt. Respond with your userid and password and you should receive a shell prompt, at which you can enter commands and view their output.

If you have only a single phone line, you'll be unable to place a call to your Linux system. You'll have to call from some other location.



Learning Debian GNU/Linux

By [Bill McCarty](#)

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

13. Conquering the BASH Shell

Contents:

[The Linux Shell](#)

[Using the Shell](#)

[Understanding Shell Scripts](#)

This chapter describes the powerful BASH shell, providing a much more detailed explanation than that provided in **Chapter 4**,

Issuing Linux Commands. The chapter also briefly explains shell variables, shell scripts, and shell aliases, preparing you for an in-depth, continuing study of Linux.

13.1 The Linux Shell

You met the Linux command interpreter, or shell, early in this book. Like an MS-DOS Prompt window, the shell lets you issue commands that it interprets, or executes. By means of the shell, you use and control your system.

13.1.1 A Variety of Shells

The MS-DOS shell has been fairly consistent over time; for example, the differences between MS-DOS v3 and MS-DOS v7 are few. The Unix shell, however, has experienced significantly more evolutionary development than MS-DOS. Today, you find both versions and variants of the Unix shell. The Unix shell variants have much in common, but each has a different authorship and history, and each reflects a different view of how users should interact with Unix.

Linux includes the most popular Unix shells, as shown in [Table 13.1](#). The most popular Linux shell is the BASH shell (the "Bourne Again SHell"), based on the original Unix Bourne shell. The BASH shell is largely compliant with the POSIX standard, which specifies the syntax and operation of a standard Unix shell and which has been widely implemented. Because of the popularity of the POSIX standard and the obvious advantage of working with a shell that's consistent across a variety of computing platforms, this chapter focuses on the BASH shell. Most Linux systems are configured to automatically start a BASH shell on your behalf when you log in; so, you don't generally need to be much concerned about choosing a shell. However, for information about the other available shells, you can consult the Linux man pages.

Table 13.1: Common Linux Shells

Shell name	Program name(s)	Description
ASH shell	<i>/bin/ash</i> <i>/bin/bsh</i>	Resembles the shell used by AT&T's System V Unix.

Shell name	Program name(s)	Description
BASH shell	<i>/bin/bash</i> <i>/bin/bash2</i>	The standard shell for Linux, based on the original Unix Bourne shell. According to its man page, BASH is "ultimately intended" to be POSIX compliant.
C shell	<i>/bin/csh</i> <i>/bin/tcsh</i>	The second Unix shell. Designed to facilitate interactive use, it added many new features and functions. Its syntax resembles that of the C programming language.
Korn shell	<i>/bin/ksh</i>	The third Unix shell, added many of the features of the C shell to the original Bourne shell.
Z shell	<i>/bin/zsh</i>	A feature-packed shell based on the Korn shell.

13.1.2 Why Learn to Use the Shell?

If you're accustomed to the point-and-click world of graphical user interfaces, you may question the value of learning to use the Linux shell. Many users initially find the shell cumbersome, and some retreat to the familiar comfort of the graphical user interface, avoiding the shell whenever possible.

While it's true that the shell is an older style of interacting with a computer than the graphical user interface, the graphical user interface is actually the more primitive interface. The graphical user interface is easy to learn and widely used, but the shell is vastly more sophisticated. Using a graphical user interface is somewhat like communicating in American Indian sign language. If your message is a simple one, like "we come in peace," you can communicate it by using a few gestures. However, if you attempted to give Lincoln's Gettysburg address - a notably short public discourse - by means of American Indian sign language, you'd find your task quite formidable.

American Sign Language, used to communicate with those who have a hearing impairment, is a much richer language than American Indian sign language. Unfortunately, programmers have not yet risen to the challenge of creating graphical user interfaces that are equally sophisticated. The designer of a program

that provides a graphical user interface must anticipate all the possible ways in which the user will interact with the program and provide ways to trigger the appropriate program responses by means of pointing and clicking. Consequently, the user is constrained to working only in predicted ways. The user is therefore unable to adapt the graphical user interface program to accommodate unforeseen tasks and circumstances. In a nutshell, that's why many system administration tasks are performed using the shell: system administrators, in fulfilling their responsibility to keep a system up and running, must continually deal with and overcome the unforeseen.

The shell reflects the underlying philosophy of Unix, which provides a wide variety of small, simple tools (that is, programs), each performing a single task. When a complex operation is needed, the tools work together to accomplish the complex operation as a series of simple operations, one step at a time. Many Unix tools manipulate text and, since Unix stores its configuration data in text form rather than in binary form, the tools are ideally suited for manipulating Unix itself. The shell's ability to freely combine tools in novel ways is what makes Unix powerful and sophisticated. Moreover, as you'll learn, the shell is extensible: You can create shell scripts that let you store a series of commands for later execution, saving you the future tedium of typing or pointing and clicking to recall them.

The contrary philosophy is seen in operating systems such as Microsoft Windows, which employ elaborate, monolithic programs that provide menus, submenus, and dialog boxes. Such programs have no way to cooperate with one another to accomplish complex operations that weren't anticipated when the programs were designed. They're easy to use so long as you remain on the beaten path, but once you step off the trail you find yourself in a confusing wilderness.

Of course, not everyone shares this perspective. The USENET newsgroups, for example, are filled with postings debating the relative merits of graphical user interfaces. Some see the Unix shell as an arcane and intimidating monstrosity. But, even if they're correct, it's inarguable that when you learn to use the shell, you begin to see Unix as it was intended (whether that's for better or for worse).

The author's perspective is pragmatic: When performing common, routine operations, a graphical user interface that minimizes typing can be a relief; but, when faced with a complex, unstructured problem that requires creative solution, the shell is more often the tool of choice. By creating solutions in the form of shell scripts, solutions can be stored for subsequent reuse. Perhaps even more important, shell scripts can be studied to quickly bone up on forgotten details, expediting the solution of related problems.

13.2 Using the Shell

This book introduced you to the shell in **Chapter 4**. However, many important details were omitted in that chapter, which was aimed at helping you to get your Linux system up and running as quickly as possible. This section revisits the shell, providing you with information that will help you use the shell efficiently and effectively.

13.2.1 Typing Shell Commands

When typing shell commands, you have access to a mini-editor that resembles the DOSKEYS editor of MS-DOS. [Table 13.2](#) summarizes some useful keystroke commands interpreted by the shell. The keystroke commands let you access a list of recently executed commands, called the *history list*. To re-execute a command, you can press the Up key several times until you locate the command and then merely press **Enter** to execute the command.

Table 13.2: Useful Editing Keystrokes

Keystroke(s)	Function
Up	Move back one command in the history list.
Down	Move forward one command in the history list.
Left	Move back one character.
Right	Move forward one character.
Esc f	Move forward one word.
Esc b	Move back one word.
Ctrl-A	Move to beginning of line.

Keystroke(s)	Function
Ctrl-E	Move to end of line.
Ctrl-D	Delete current character.
Backspace	Delete previous character.
Esc d	Delete current word.
Ctrl-U	Delete from beginning of line.
Esc k	Delete to end of line.
Ctrl-Y	Retrieve last item deleted.
Esc .	Insert last word of previous command.
Ctrl-L	Clear the screen, placing the current line at the top of the screen.
Tab	Attempt to complete the current word, interpreting it as a filename, username, variable name, hostname, or command as determined by the context.
Esc ?	List the possible completions.

One of the most useful editing keystrokes, **Tab**, can also be used when typing a command. If you type the first part of a filename and press **Tab**, the shell will

attempt to locate files with names matching the characters you've typed. If exactly one such file exists, the shell fills out the partially typed name with the proper characters. You can then press **Enter** to execute the command or continue typing other options and arguments. This feature, called either filename completion or command completion, makes the shell much easier to use.

In addition to keystrokes for editing the command line, the shell interprets several keystrokes that control the operation of the currently executing program. [Table 13.3](#) summarizes these keystrokes. For example, typing Ctrl-C generally cancels execution of a program. This keystroke command is handy, for example, when a program is taking too long to execute and you'd prefer to try something else.

Table 13.3: Useful Control Keystrokes

Keystroke	Function
Ctrl-C	Sends an interrupt signal to the currently executing command, which generally responds by terminating itself.
Ctrl-D	Sends an end of file to the currently executing command. Use this keystroke to terminate console input.
Ctrl-Z	Suspends the currently executing program.

Several other special characters control the operation of the shell, as shown in [Table 13.4](#). The # and ; characters are most often used in shell scripts, which you'll learn about later in this chapter. The & character is useful for running a command as a background process.

Table 13.4: Other Special Shell Characters

Character	Function
#	Marks the command as a comment, which the shell ignores.

Character	Function
;	Separates commands, letting you enter several commands on a single line.
&	Placed at the end of a command, causes the command to execute as a background process, so that a new shell prompt appears immediately after the command is entered.

13.2.2 Commands and Arguments

As you already know, the general form a shell command line is this:

```
command options arguments
```

The command determines what operation the shell will perform and the options and arguments customize or fine-tune the operation. Sometimes the command specifies a program file that will be launched and run; such a command is called an *external command*. Linux generally stores these files in */bin*, */usr/bin*, or */usr/local/bin*. System administration commands are generally stored in */sbin* or */usr/sbin*. When a command specifies a program file, the shell passes any specified arguments to the program, which scans them and interprets them, adjusting its operation accordingly.

However, some commands are not program files; instead they are built-in commands interpreted by the shell itself. One important way in which shells differ is the built-in commands that they support. Later in this section, you'll learn about some commands built into the BASH shell.

13.2.3 Filename Globbing

Before the shell passes arguments to an external command or interprets a built-in command, it scans the command line for certain special characters and performs an operation known as *filename globbing*. Filename globbing resembles the processing of wildcards used in MS-DOS commands, but it's much more sophisticated. [Table 13.5](#) describes the special characters used in filename globbing, known as *filename metacharacters*.

Table 13.5: Filename Metacharacters

--

Metacharacter	Meaning
*	Matches a string of zero or more characters
?	Matches exactly one character
[<i>abc</i> ...]	Matches any of the characters specified
[<i>a - z</i>]	Matches any character in the specified range
[! <i>abc</i> ...]	Matches any character other than those specified
[! <i>a - z</i>]	Matches any character not in the specified range
~	The home directory of the current user
~ <i>userid</i>	The home directory of the specified user
~+	The current working directory
~-	The previous working directory

In filename globbing just as in MS-DOS wildcarding, the shell attempts to replace metacharacters appearing in arguments in such a way that arguments specify filenames. Filename globbing makes it easier to specify names of files and sets of files.

For example, suppose the current working directory contains the following files: *file1*, *file2*, *file3*, and *file04*. Suppose you want to know the size of each file. The following command reports that information:

```
ls -l file1 file2 file3 file04
```


However, the following command reports the same information and is much easier to type:

```
ls -l file*
```

As [Table 13.2](#) shows, the `*` filename metacharacter can match any string of characters. Suppose you issued the following command:

```
ls -l file?
```

The `?` filename metacharacter can match only a single character. Therefore, *file04* would not appear in the output of the command.

Similarly, the command:

```
ls -l file[2-3]
```

would report only *file2* and *file3*, because only these files have names that match the specified pattern, which requires that the last character of the filename be in the range 2-3.

You can use more than one metacharacter in a single argument. For example, consider the following command:

```
ls -l file??
```

This command will list *file04*, because each metacharacter matches exactly one filename character.

Most commands let you specify multiple arguments. If no files match a given argument, the command ignores the argument. Here's another command that reports all four files:

```
ls -l file0* file[1-3]
```

Suppose that a command has one or more arguments that include one or more metacharacters. If none of the arguments matches any filenames, the shell passes the arguments to the program with the metacharacters intact. When the program expects a valid filename, an unexpected error may result.

Another metacharacter lets you easily refer to your home directory. For example, the following command:

```
ls ~
```

lists the files in the user's home directory.

Filename metacharacters don't merely save you typing. They let you write scripts that selectively process files by name. You'll see how that works later in this chapter.

13.2.4 Shell Aliases

Shell aliases make it easier to use commands by letting you establish abbreviated command names and by letting you pre-specify common arguments. To establish a command alias, issue a command of the form:

```
alias
name='
command'
```

where *command* specifies the command for which you want to create an alias and *name* specifies the alias. For example, suppose you frequently type the MS-DOS command `Dir` when you intend to type the Linux command `ls`. You can establish an alias for the `ls` command by issuing this command:

```
alias dir='ls -l'
```

Once the alias is established, if you mistakenly type `Dir`, you'll nevertheless get the directory listing you want. If you like, you can establish similar aliases for other commands.

Your default Linux configuration probably defines several aliases on your behalf. To see what they are, issue the command:

```
alias
```

If you're logged in as `root`, you may see the following aliases:

```
alias cp='cp -i'
alias dir='ls -l'
alias ls='ls --color'
alias mv='mv -i'
alias rm='rm -i'
```

Notice how several commands are self-aliased. For example, the command `rm -i` is aliased as `rm`. The effect is that the `-i` option appears whenever you issue the `rm` command, whether or not you type the option. The `-i` option specifies that the shell will prompt for confirmation before deleting files. This helps avoid accidental deletion of files, which can be particularly hazardous when you're logged in as `root`. The alias ensures that you're prompted for confirmation even if you don't ask to be prompted. If you don't want to be prompted, you can issue a command like:

```
rm -f
files
```

where *files* specifies the files to be deleted. The *-f* option has an effect opposite that of the *-i* option; it forces deletion of files without prompting for confirmation. Because the command is aliased, the command actually executed is:

```
rm -i -f
files
```

The *-f* option takes precedence over the *-i* option, because it occurs later in the command line.

If you want to remove a command alias, you can issue the `unalias` command:

```
unalias
alias
```

where *alias* specifies the alias you want to remove. Aliases last only for the duration of a log in session, so you needn't bother to remove them before logging off. If you want an alias to be effective each time you log in, you can use a shell script. The next subsection shows you how to do so.

13.2.5 Shell Scripts

A shell script is simply a file that contains commands. By storing commands as a shell script you make it easy to execute them again and again. As an example, consider a file named *deleter*, which contains the following lines:

```
echo -n Deleting the temporary files...
rm -f *.tmp
echo Done.
```

The `echo` commands simply print text on the console. The *-n* option of the first `echo` command causes omission of the trailing newline character normally written by the `echo` command, so both `echo` commands write their text on a single line. The `rm` command removes from the current working directory all files having names ending in *.tmp*.

You can execute this script by issuing the `sh` command:

```
sh deleter
```

If you invoke the `sh` command without an argument specifying a script file, a new interactive shell is launched. To exit the new shell and return to your previous session, issue the `exit` command.

If the *deleter* file were in a directory other than the current working directory, you'd have to type an absolute path, for example:

```
sh /home/bill/deleter
```

You can make it a bit easier to execute the script by changing its access mode to include execute access. To do so, issue the following command:

```
chmod ugo+x deleter
```

This gives you, members of your group, and everyone else the ability to execute the file. To do so, simply type the absolute path of the file, for example:

```
/home/bill/deleter
```

If the file is in the current directory, you can issue the following command:

```
./deleter
```

You may wonder why you can't simply issue the command:

```
deleter
```

In fact, this still simpler form of the command will work, so long as *deleter* resides in a directory on your search path. You'll learn about the search path later.

Linux includes several standard scripts that are run at various times. [Table 13.6](#) identifies these and gives the time when each is run. You can modify these scripts to operate differently. For example, if you want to establish command aliases that are available whenever you log in, you can use a text editor to add the appropriate lines to the *.profile* file that resides in your home directory. Recall that, since the name of this file begins with a dot, the `ls` command won't normally show the file. You must specify the `-a` option in order to see this and other hidden files.

Table 13.6: Special Scripts

Script	Function
<i>/etc/profile</i>	Executed when the user logs in
<i>~/.profile</i>	Executed when the user logs in

Script	Function
<code>~/.bashrc</code>	Executed when BASH is launched
<code>~/.bash_logout</code>	Executed when the user logs out

If you want to modify one of the standard scripts that should reside in your home directory, but find that your home directory does not contain the indicated file, simply create the file. The next time you log in, log out, or launch BASH (as appropriate) the shell will execute your script.

13.2.6 Input/Output Redirection and Piping

The shell provides three standard data streams:

`stdin`

The standard input stream

`stdout`

The standard output stream

`stderr`

The standard error stream

By default, most programs read their input from `stdin` and write their output to `stdout`. Because both streams are normally associated with a console, programs behave as you generally want, reading input data from the console keyboard and writing output to the console screen. When a well-behaved program writes an error message, it writes the message to the `stderr` stream, which is also associated with the console by default. Having separate streams for output and error messages presents an important opportunity, as you'll see in a moment.

Although the shell associates the three standard input/output streams with the console by default, you can specify input/output redirectors that, for example, associate an input or output stream with a file. [Table 13.7](#) summarizes the most important input/output redirectors.

Table 13.7: Input/Output Redirectors

Redirector	Function
<code>> file</code>	Redirects standard output stream to specified file
<code>2> file</code>	Redirects standard error stream to specified file
<code>>> file</code>	Redirects standard output stream to specified file, appending output to the file if the file already exists
<code>2>> file</code>	Redirects standard error stream to specified file, appending output to the file if the file already exists
<code>&> file</code>	Redirects standard output and error streams to the specified file
<code>< file</code>	Redirects standard input stream to the specified file
<code><< text</code>	Reads standard input until a line matching <i>text</i> is found, at which point end of file is posted
<code>cmd1 cmd2</code>	Takes the standard input of <i>cmd2</i> from the standard output of <i>cmd1</i> (also known as the <i>pipe redirector</i>)

To see how redirection works, consider the `wc` command. This command takes a series of filenames as arguments and prints the total number of lines, words, and characters present in the specified files. For example, the command:

```
wc /etc/passwd
```

might produce the output:

```
22      26      790 /etc/passwd
```

which indicates that the file `/etc/passwd` contains 22 lines, 26 words, and 790 characters. Generally, the output of the command appears on console. But, consider the following command, which includes an output redirector:

```
wc /etc/passwd > total
```

If you issue this command, you'll see no console output, because the output is redirected to the file *total*, which the command creates (or overwrites, if the file already exists). If you execute the pair of commands:

```
wc /etc/passwd > total
cat total
```

you can see the output of the `wc` command on the console.

Perhaps you can now see the reason for having the separate output streams `stdout` and `stderr`. If the shell provided a single output stream, error messages and output would be mingled. Therefore, if you redirected the output of a program to a file, any error messages would also be redirected to the file. This might make it difficult to notice an error that occurred during program execution. Instead, because the streams are separate, you can choose to redirect only `stdout` to a file. When you do so, error messages sent to `stderr` appear on the console in the usual way. Of course, if you prefer, you can redirect both `stdout` and `stderr` to the same file or redirect them to different files. As usual in the Unix world, you can have it your own way.

A simple way of avoiding annoying output is to redirect it to the null file, */dev/null*. If you redirect the `stderr` stream of a command to */dev/null*, you won't see any error messages the command produces.

Just as you can direct the standard output or error stream of a command to a file, you can also redirect a command's standard input stream to a file, so that the command reads from the file instead of the console. For example, if you issue the `wc` command without arguments, the command reads its input from `stdin`. Type some words and then type the end of file character (Ctrl-D) and `wc` will report the number of lines, words, and characters you entered. You can tell `wc` to read from a file, rather than the console, by issuing a command like:

```
wc </etc/passwd
```

Of course, this isn't the usual way of invoking `wc`. The author of `wc` helpfully provided a command-line argument that lets you specify the file from which `wc` reads. However, by using a redirector, you could read from any desired file even if the author had been less helpful.

Some programs are written to ignore redirectors. For example, the `passwd` command expects to read the new password only from the console, not from a file. You can compel such programs to read from a file, but doing so requires techniques more advanced than redirectors.

When you specify no command-line arguments, many Unix programs read their input from `stdin` and write their output to `stdout`. Such programs are called *filters*. Filters can be easily fitted together to perform a series of related operations. The tool for combining filters is the *pipe*, which connects the output of one program to the input of another. For example, consider this command:

```
ls -l ~ | wc -l
```

The command consists of two commands, joined by the pipe redirector (`|`). The first command lists the names of the files in the user's home directory, one file per line. The second command invokes `wc` by using the `-l` option, which causes `wc` to print only the total number of lines, rather than printing the total number of lines, words, and characters. The pipe redirector sends the output of the `ls` command to the `wc` command, which counts and prints the number of lines in its input, which happens to be the number of files in the user's home directory.

This is a simple example of the power and sophistication of the Unix shell. Unix doesn't include a command that counts the files in the user's home directory and doesn't need to do so. Should the need to count the files arise, a knowledgeable Unix user can prepare a simple script that computes the desired result by using general-purpose Unix commands.

13.2.7 Shell Variables

If you've studied programming, you know that programming languages resemble algebra. Both programming languages and algebra let you refer to a value by a name. And both programming languages and algebra include elaborate mechanisms for manipulating named values.

The shell is a programming language in its own right, letting you refer to variables known as *shell variables* or *environment variables*. To assign a value to a shell variable, you use a command that has the following form:

```
variable=  
value
```

For example, the command:

```
DifficultyLevel=1
```

assigns the value `1` to the shell variable named `DifficultyLevel`. Unlike algebraic variable, shell variables can have non-numeric values. For example, the command:

```
Difficulty=medium
```


assigns the value `medium` to the shell variable named `Difficulty`.

Shell variables are widely used within Unix, because they provide a convenient way of transferring values from one command to another. Programs can obtain the value of a shell variable and use the value to modify their operation, in much the same way they use the value of command-line arguments.

You can see a list of shell variables by issuing the `set` command. Usually, the command produces more than a single screen of output. So, you can use a pipe redirector and the `more` command to view the output one screen at a time:

```
set | more
```

Press the **Space** bar to see each successive page of output. You'll probably see several of the shell variables described in [Table 13.8](#) among those printed by the `set` command. The values of these shell variables are generally set by one or another of the startup scripts described earlier in this chapter.

Table 13.8: Important Environment Variables

Variable	Function
DISPLAY	The X display to be used; for example, <code>localhost:0</code>
HOME	The absolute path of the user's home directory
HOSTNAME	The Internet name of the host
LOGNAME	The user's login name
MAIL	The absolute path of the user's mail file
PATH	The search path (see next subsection)
SHELL	The absolute path of the current shell

Variable	Function
TERM	The terminal type
USER	The user's current username; may differ from the login name if the user executes the <code>su</code> command

You can use the value of a shell variable in a command by preceding the name of the shell variable by a dollar sign (\$). To avoid confusion with surrounding text, you can enclose the name of the shell variable within curly braces ({}); it's good practice (though not necessary) to do so consistently. For example, you can change the current working directory to your home directory by issuing the command:

```
cd ${HOME}
```

Of course, issuing the `cd` command with no argument causes the same result. However, suppose you want to change to the *work* subdirectory of your home directory. The following command accomplishes exactly that:

```
cd ${HOME}/work
```

An easy way to see the value of a shell variable is to specify the variable as the argument of the `echo` command. For example, to see the value of the `HOME` shell variable, issue the command:

```
echo ${HOME}
```

To make the value of a shell variable available not just to the shell, but to programs invoked by using the shell, you must export the shell variable. To do so, use the `export` command, which has the form:

```
export
variable
```

where *variable* specifies the name of the variable to be exported. A shorthand form of the command lets you assign a value to a shell variable and export the variable in a single command:

```
export
variable=
value
```

You can remove the value associated with shell variable by giving the variable an empty value:

```
variable=
```

However, a shell variable with an empty value remains a shell variable and appears in the output of the `set` command. To dispense with a shell variable, you can issue the `unset` command:

```
unset  
variable
```

Once you unset the value of a variable, the variable no longer appears in the output of the `set` command.

13.2.8 The Search Path

The special shell variable `PATH` holds a series of paths known collectively as the *search path*. Whenever you issue an external command, the shell searches paths that comprise the search path, seeking the program file that corresponds to the command. The startup scripts establish the initial value of the `PATH` shell variable, but you can modify its value to include any desired series of paths. You must use a colon (:) to separate each path of the search path.

For example, suppose that `PATH` has the following value:

```
/usr/bin:/bin:/usr/local/bin:/usr/bin/X11:/usr/X11R6/bin
```

You can add a new search directory, say */home/bill*, with the following command:

```
PATH=${PATH}:/home/bill
```

Now, the shell will look for external programs in */home/bill* as well as the default directories. However, it will look there last. If you prefer to check */home/bill* first, issue the following command instead:

```
PATH=/home/bill:${PATH}
```

The `which` command helps you work with the `PATH` shell variable. It checks the search path for the file specified as its argument and prints the name of the matching path, if any. For example, suppose you want to know where the program file for the `wc` command resides. Issuing the command:

```
which wc
```

will tell you that the program file is `/usr/bin/wc`, or whatever other path is correct for your system.

13.2.9 Quoted Strings

Sometimes the shell may misinterpret a command that you've written, globbing a filename or expanding a reference to a shell variable that you hadn't intended. Of course, it's actually your interpretation that's mistaken, not the shell's. Therefore, it's up to you to rewrite your command so that the shell's interpretation is congruent with what you intend.

Quote characters, described in [Table 13.9](#), can help you do so, by controlling the operation of the shell. For example, by enclosing a command argument within single quotes, you can prevent the shell from globbing the argument or substituting the argument with the value of a shell variable.

Table 13.9: Quote Characters

Character	Function
'	Characters within a pair of single quotes are interpreted literally; that is, their metacharacter meanings (if any) are ignored. Similarly, the shell does not replace references to shell or environment variables with the value of the referenced variable.
"	Characters within a pair of double quotes are interpreted literally; that is, their metacharacter meanings (if any) are ignored. However, the shell does replace references to shell or environment variables with the value of the referenced variable.
`	Text within a pair of back quotes is interpreted as a command, which the shell executes before executing the rest of the command line. The output of the command replaces the original back-quoted text.

Character	Function
\	The following character is interpreted literally; that is, its metacharacter meaning (if any) is ignored. The backslash character has a special use as a line continuation character. When a line ends with a backslash, the line and the following line are considered part of a single line.

To see this in action, consider how you might cause the `echo` command to produce the output `$PATH`. If you simply issue the command:

```
echo $PATH
```

the `echo` command will print the value of the `PATH` shell variable. However, by enclosing the argument within single quotes, you obtain the desired result:

```
echo '$PATH'
```

Double quotes have a similar effect. They prevent the shell from globbing a filename but permit the expansion of shell variables.

Back quotes operate differently; they let you execute a command and use its output as an argument of another command. For example, the command:

```
echo My home directory contains `ls ~ | wc -l` files.
```

prints a message that gives the number of files in the user's home directory. The command works by first executing the command contained within back quotes:

```
ls ~ | wc -l
```

This command, as explained earlier, computes and prints the number of files in the user's directory. Because the command is enclosed in back quotes, its output is not printed; instead the output replaces the original back quoted text.

The resulting command becomes:

```
echo My home directory contains 22 files.
```

When executed, this command prints the output:

```
My home directory contains 22 files.
```

13.2.10 The Power of the Linux Shell

You may now begin to appreciate the power of the Linux shell: by including command aliases in your *bashrc* script, you can extend the command repertoire of the shell. And, by using filename completion and the history list, you can reduce the amount of typing necessary. Once you grasp how to properly use it, the Linux shell is a powerful, fast, and easy to use interface that avoids the limitations and monotony of the more familiar point-and-click graphical interface.

But, the shell has additional features that extend its capabilities even further. As you'll see in the next section, the Linux shell includes a powerful programming language that provides argument processing, conditional logic, and loops.

13.3 Understanding Shell Scripts

This section explains how more advanced shell scripts work. The information is also adequate to equip you to write many of your own useful shell scripts. The section begins by showing how to process a script's arguments. Then it shows how to perform conditional and iterative operations.

13.3.1 Processing Arguments

You can easily write scripts that process arguments, because a set of special shell variables holds the values of arguments specified when your script is invoked. [Table 13.10](#) describes the most popular such shell variables.

For example, here's a simple one-line script that prints the value of its second argument:

```
echo My second argument has the value $2.
```

Suppose you store this script in the file *second*, change its access mode to permit execution, and invoke it as follows:

```
./second a b c
```

The script will print the output:

```
My second argument has the value b.
```

Table 13.10: Special Shell Variables Used in Scripts

Variable	Meaning
\$#	The number of arguments.
\$0	The command name.
\$1, \$2, ... , \$9	The individual arguments of the command.

Variable	Meaning
\$*	The entire list of arguments, treated as a single word.
\$@	The entire list of arguments, treated as a series of words.
\$?	The exit status of the previous command. The value 0 denotes successful completion.
\$\$	The process id of the current process.

Notice that the shell provides variables for accessing only nine arguments. Nevertheless, you can access more than nine arguments. The key to doing so is the `shift` command, which discards the value of the first argument and shifts the remaining values down one position. Thus, after executing the `shift` command, the shell variable `$9` contains the value of the tenth argument. To access the eleventh and subsequent arguments, you simply execute the `shift` command the appropriate number of times.

13.3.2 Exit Codes

The shell variable `$?` holds the numeric exit status of the most recently completed command. By convention, an exit status of zero denotes successful completion; other values denote error conditions of various sorts.

You can set the error code in a script by issuing the `exit` command, which terminates the script and posts the specified exit status. The format of the command is:

```
exit
status
```

where *status* is a non-negative integer that specifies the exit status.

13.3.3 Conditional Logic

A shell script can employ conditional logic, which lets the script take different action based on the values of arguments, shell variables, or other conditions. The

`test` command lets you specify a condition, which can be either true or false. Conditional commands (including the `if`, `case`, `while`, and `until` commands) use the `test` command to evaluate conditions.

13.3.3.1 The test command

[Table 13.11](#) describes some commonly used argument forms used with the `test` command. The `test` command evaluates its arguments and sets the exit status to 0, which indicates that the specified condition was true, or a non-zero value, which indicates that the specified condition was false.

Table 13.11: Commonly Used Argument Forms of the test Command

Form	Function
<code>-d</code> <i>file</i>	The specified file exists and is a directory.
<code>-e</code> <i>file</i>	The specified file exists.
<code>-r</code> <i>file</i>	The specified file exists and is readable.
<code>-s</code> <i>file</i>	The specified file exists and has non-zero size.
<code>-w</code> <i>file</i>	The specified file exists and is writable.
<code>-x</code> <i>file</i>	The specified file exists and is executable.
<code>-L</code> <i>file</i>	The specified file exists and is a symbolic link.

Form	Function
$f1 \text{ -nt } f2$	File $f1$ is newer than file $f2$.
$f1 \text{ -ot } f2$	File $f1$ is older than file $f2$.
$\text{-n } s1$	String $s1$ has nonzero length.
$\text{-z } s1$	String $s1$ has zero length.
$s1 = s2$	String $s1$ is the same as string $s2$.
$s1 \neq s2$	String $s1$ is not the same as string $s2$.
$n1 \text{ -eq } n2$	Integer $n1$ is equal to integer $n2$.
$n1 \text{ -ge } n2$	Integer $n1$ is greater than or equal to integer $n2$.
$n1 \text{ -gt } n2$	Integer $n1$ is greater than integer $n2$.
$n1 \text{ -le } n2$	Integer $n1$ is less than integer $n2$.
$n1 \text{ -lt } n2$	Integer $n1$ is less than or equal to integer $n2$.

Form	Function
<code>n1 -ne n2</code>	Integer <i>n1</i> is not equal to integer <i>n2</i> .
<code>!</code>	The <code>not</code> operator, which reverses the value of the following condition.
<code>-a</code>	The <code>and</code> operator, which joins two conditions. Both conditions must be true for the overall result to be true.
<code>-o</code>	The <code>or</code> operator, which joins two conditions. If either condition is true, the overall result is true.
<code>\(... \)</code>	You can group expressions within the <code>test</code> command by enclosing them within <code>\(</code> and <code>\)</code> .

To see the `test` command in action, consider the following script:

```
test -d $1
echo $?
```

This script tests whether its first argument specifies a directory and displays the resulting exit status, a zero or a non-zero value that reflects the result of the test.

Suppose the script were stored in the file *tester*, which permitted read access. Executing the script might yield results similar to the following:

```
$ ./tester /
0
$ ./tester /missing
1
```

These results indicate that the `/` directory exists and that the `/missing` directory does not exist.

13.3.3.2 The `if` command

The `test` command is not of much use by itself, but combined with commands such as the `if` command, it is useful indeed. The `if` command has the following

form:

```
if
command
then

commands
else

commands
fi
```

Usually the command that immediately follows the word `if` is a `test` command. However, this need not be so. The `if` command merely executes the specified command and tests its exit status. If the exit status is 0, the first set of commands is executed; otherwise the second set of commands is executed. An abbreviated form of the `if` command does nothing if the specified condition is false:

```
if
command
then

commands
fi
```

When you type an `if` command, it occupies several lines; nevertheless it's considered a single command. To underscore this, the shell provides a special prompt (called the *secondary prompt*) after you enter each line. Often, scripts are entered by using a text editor; when you enter a script using a text editor you don't see the secondary prompt, or any other shell prompt for that matter.

As an example, suppose you want to delete a file *file1* if it's older than another file *file2*. The following command would accomplish the desired result:

```
if test file1 -ot file2
then
    rm file1
fi
```

You could incorporate this command in a script that accepts arguments specifying the filenames:

```
if test $1 -ot $2
then
    rm $1
    echo Deleted the old file.
fi
```

If you name the script *riddance* and invoke it as follows:

```
riddance thursday wednesday
```

the script will delete the file *thursday* if that file is older than the file *wednesday*.

13.3.3.3 The case command

The `case` command provides a more sophisticated form of conditional processing:

```
case
value in

pattern1)
commands;;

pattern2)
commands ;;

...
esac
```

The `case` command attempts to match the specified value against a series of patterns. The commands associated with the first matching pattern, if any, are executed. Patterns are built using characters and metacharacters, such as those used to specify command arguments. As an example, here's a `case` command that interprets the value of the first argument of its script:

```
case $1 in
-r) echo Force deletion without confirmation ;;
-i) echo Confirm before deleting ;;
*) echo Unknown argument ;;
esac
```

The command echoes a different line of text, depending on the value of the script's first argument. As done here, it's good practice to include a final pattern that matches any value.

13.3.3.4 The while command

The `while` command lets you execute a series of commands iteratively (that is, repeatedly) so long as a condition tests true:

```
while
command
do

commands
done
```

Here's a script that uses a `while` command to print its arguments on successive lines:

```
echo $1
while shift 2> /dev/null
do
    echo $1
done
```

The commands that comprise the `do` part of a `while` (or another loop command) can include `if` commands, `case` commands, and even other `while` commands. However, scripts rapidly become difficult to understand when this occurs often. You should include conditional commands within other conditional commands only with due consideration for the clarity of the result. Include a comment command (`#`) to clarify difficult constructs.

13.3.3.5 The `until` command

The `until` command lets you execute a series of commands iteratively (that is, repeatedly) so long as a condition tests false:

```
until
command
do

    commands
done
```

Here's a script that uses an `until` command to print its arguments on successive lines, until it encounters an argument that has the value *red*:

```
until test $1 = red
do
    echo $1
    shift
done
```

For example, if the script were named *stopandgo* and stored in the current working directory, the command:

```
./stopandgo green yellow red blue
```

would print the lines:

```
green
yellow
```

13.3.3.6 The `for` command

The `for` command iterates over the elements of a specified list:

```
for
variable in
list
do

commands
done
```

Within the commands, you can reference the current element of the list by means of the shell variable `$ variable`, where `variable` is the name specified following the `for`. The list typically takes the form of a series of arguments, which can incorporate metacharacters. For example, the following `for` command:

```
for i in 2 4 6 8
do
    echo $i
done
```

prints the numbers 2, 4, 6, and 8 on successive lines.

A special form of the `for` command iterates over the arguments of a script:

```
for
variable
do

commands
done
```

For example, the following script prints its arguments on successive lines:

```
for i
do
    echo $i
done
```

13.3.3.7 The `break` and `continue` commands

The `break` and `continue` commands are simple commands that take no arguments. When the shell encounters a `break` command, it immediately exits the body of the enclosing loop (`while`, `until`, or `for`) command. When the shell encounters a `continue` command, it immediately discontinues the current iteration of the loop. If the loop condition permits, other iterations may occur; otherwise the loop is exited.

13.3.4 Periscope: A Useful Networking Script

Suppose you have a free email account such as that provided by Yahoo! You're traveling and find yourself in a remote location with Web access. However, you're unable to access files on your home machine or check email that has arrived there. This is a common circumstance, especially if your business requires that you travel.

If your home computer runs Microsoft Windows, you're pretty much out of luck. You'll find it extraordinarily difficult to access your home computer from afar. However, if your home computer runs Linux, gaining access is practically a piece of cake.

In order to show the power of shell scripts, this subsection explains a more complex shell script, `periscope`. At an appointed time each day, `periscope` causes your computer (which you must leave powered on) to establish a PPP connection to your ISP, which is maintained for about one hour. This provides you enough time to connect to an ISP from your hotel room or other remote location and then connect via the Internet with your home Linux system, avoiding long distance charges. Once connected, you have about an hour to view or download mail and perform other work. Then, `periscope` breaks its PPP connection, which it will re-establish at the appointed time the next day.

[Example 13.1](#) shows the `periscope` script file, which is considerably larger than any script you've so far encountered in this chapter. Therefore, we'll disassemble the script, explaining it line by line. As you'll see, each line is fairly simple in itself and the lines work together in a straightforward fashion.

Example 13.1: Periscope

```
PATH=${PATH}:/usr/local/bin
route del default
wvdial &
sleep 1m
ifconfig | mail
userid@mail.com
sleep 1h
killall wvdial
sleep 2s
killall -9 wvdial
killall pppd
sleep 2s
killall -9 pppd
echo "/root/periscope" | at 10:00
```

The first line of the script augments the search path for the script to include `/usr/local/bin`, the directory that contains the `wvdial` external command. Some versions of the startup scripts may not include this path in the search path, so explicitly placing it there avoids a possible problem.


```
PATH=${PATH}:/usr/local/bin
```

The next line is perhaps the most complex line of the entire script:

```
route del default
```

The `route` command is normally issued by the system administrator. You've probably never issued the command yourself, because a network configuration program has issued it on your behalf. The effect of the command is to delete the default network route, if any. The default route is the one along which TCP/IP sends packets when it knows no specific route to their specified destination. It's necessary to delete the default route because the `wvdial` program, which the script uses to establish its PPP connection, will not override an existing default route.

```
wvdial &
```

The next line launches the `wvdial` program. As specified by the ampersand (&), the program runs in the background, so the script continues executing while `wvdial` starts up and runs. The next line pauses the script for one minute, giving `wvdial` time to establish the PPP connection:

```
sleep 1m
```

The next line runs the `ifconfig` command and mails its output to the specified user (you must replace *userid@mail.com* with your own email address, which you can access remotely):

```
ifconfig | mail  
userid@mail.com
```

The `ifconfig` command produces output that looks something like this:

```
ppp0    Link encap:Point-Point Protocol  
        inet addr:10.144.153.105  P-t-P:10.144.153.52 Mask:255.255.255.0  
        UP POINTOPOINT RUNNING  MTU:552  Metric:1  
        RX packets:0 errors:0 dropped:0 overruns:0  
        TX packets:0 errors:0 dropped:0 overruns:0
```

You'll probably see other sections describing your Ethernet interface (*eth0*) and a loopback device (*lo*). The `inet addr` given in the command output (10.144.153.105) is the IP address of your computer. By mailing the output to yourself, you provide a simple way to discover your computer's IP address, which is likely to be different each time it connects to your ISP.

The next line causes the script to pause for an interval of one hour:

```
sleep 1h
```

You can easily change this interval to something more appropriate to your own needs.

The connection interval now having elapsed, the next line terminates all executing instances of the `wvdial` program:

```
killall wvdial
```

Appendix E, *Linux Command Quick Reference*, briefly describes the `killall` command and other possibly unfamiliar commands employed in this script.

The script then pauses for two seconds, to ensure that `wvdial` has completely terminated:

```
sleep 2s
```

Under some circumstances, a program will ignore a termination request. The next line deals with this possibility by sending a special code that compels a reluctant program to terminate without further delay:

```
killall -9 wvdial
```

Behind the scenes, `wvdial` launches a program known as `pppd`, which actually establishes and manages the PPP connection. Another `killall` command is designed to terminate `pppd` if `wvdial` has failed to do so:

```
killall pppd
```

Again, the script pauses for a few seconds:

```
sleep 2s
```

And, again the script uses the `-9` option to specify that any remaining instances of `pppd` should terminate immediately:

```
killall -9 pppd
```

Finally, the script uses the `at` command to schedule itself for execution at 10:00 tomorrow:

```
echo "/root/periscope" | at 10:00
```

The `at` command reads one or more commands from its standard input and executes them at the time specified as an argument.

To try the script for yourself, you must have installed the `wvdial` program, as explained in **Chapter 11**,

Getting Connected to the Internet. Place the script in the file `/root/periscope`. Of course, you'll probably want to customize the script to specify an appointment time and duration of your own choosing. To start `periscope`, log in as `root` and issue the command:

```
(echo "/root/periscope" | at 10:00)&
```

When 10:00 (or such other time as you specified) comes around, your Linux system should obediently dial your ISP and maintain the connection for the specified interval of time.

13.3.5 Using `periscope`

At the appointed time, fire up your computer and access your email account. You should find a mail message that contains the `ifconfig` output giving your computer's current IP address. Now you can use `telnet` or an `ssh` client - your choice corresponds to the server you're running on your Linux system - to contact your computer and work for the remainder of the specified connection time. At the end of the connection time, your Linux system will sever its PPP connection and begin counting down until it's again time to connect.

13.3.6 Continuing Onward

Because it's quite a simple script, `periscope` doesn't do full justice to the capabilities of Linux. For example, suppose you want to establish connections at varying times or on varying days of the week. Or, suppose you want to schedule the next connection each time you log in.

Linux is able to answer such challenges in a variety of ways. For example, the `cron` program, though more complicated to use than the `at` command, provides the ability to specify program launch times very flexibly. For example, `cron` can let you establish a connection at 10:00 in the morning of the third Friday of each month.

You can learn more about Linux from **Appendix E**, which summarizes many useful Linux commands that you can use and include in shell scripts. A good way to continue learning about Linux is to peruse **Appendix E** and try each of the commands described there. Read their man pages and learn more about them. Ask yourself how the commands might be used in scripts that would facilitate your use of Linux.

If you truly catch the Linux bug, as many have, you'll want to peruse other Linux works, such as:

- *Running Linux*, 3rd edition, by Matt Welsh, Kalle Dalheimer, and Lar Kaufman (Sebastopol, CA: O'Reilly & Associates, 1999).
- *Learning the bash Shell*, 2nd edition, by Cameron Newham and Bill Rosenblatt (Sebastopol, CA: O'Reilly & Associates, 1998).
- *Linux Network Administrator's Guide*, by Olaf Kirch (Sebastopol, CA: O'Reilly & Associates, 1995).
- *Learning the vi Editor*, 6th edition, by Linda Lamb and Arnold Robbins (Sebastopol, CA: O'Reilly & Associates, 1998).

You'll also find a wealth of useful information on the Web sites described in **Chapter 1**,

Why Run Linux? and in periodicals such as *Linux Journal* and *Linux Magazine*.

However, don't merely read about Linux; work with it. Write, test, and debug your own scripts. Share scripts you've written with others and study scripts written by others. Above all, read, communicate, and share what you've learned and what you want to learn. These activities are the foundation of the Linux culture and they are means whereby Linux users - and Linux itself - grow and develop.



Learning Debian GNU/Linux

By Bill McCarty

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

A. Linux Directory Tree

Figure A.1: The Debian GNU/Linux directory tree

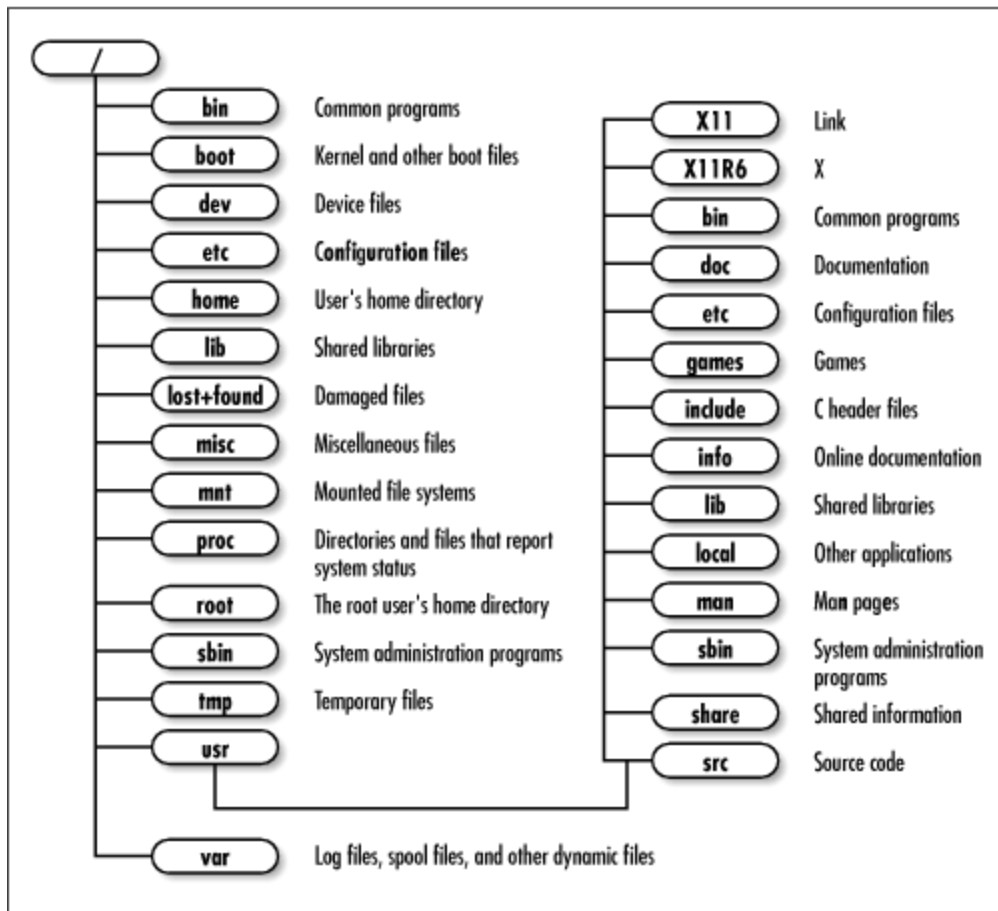
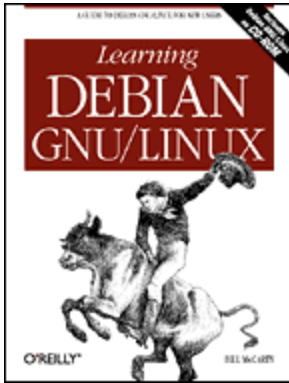


Figure A.1 shows the typical structure of the Linux directory tree. Only the principal directories are shown.



Learning Debian GNU/Linux

By Bill McCarty

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

B. Principal Linux Files

Table B.1 describes the principal Linux files. You can use it, for example, to help you locate configuration files quickly.

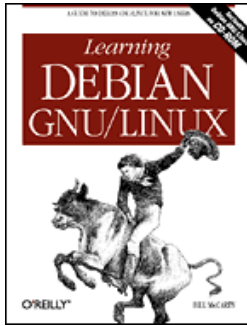
Table B.1: Principal Linux Files

File	Description
<i>/boot/vmlinuz*</i>	Linux kernel
<i>/etc/smb.conf</i>	Configuration of <code>smb</code> (Samba) daemon
<i>/etc/smbpasswd</i>	Account information for Samba users
<i>/etc/apache/access.conf</i>	Web server configuration file
<i>/etc/apache/apache.conf</i>	Web server configuration file
<i>/etc/apache/srm.conf</i>	Web server configuration file
<i>/etc/conf.modules</i>	Aliases and options for loadable kernel modules

File	Description
<i>/etc/fstab</i>	Filesystems mounted or available for mounting
<i>/etc/group</i>	Group information
<i>/etc/hosts</i>	Map of IP numbers to hostnames
<i>/etc/hosts.allow</i>	Hosts allowed to access Internet services
<i>/etc/hosts.deny</i>	Hosts forbidden to access Internet services
<i>/etc/inetd.conf</i>	Configuration for the <code>inetd</code> daemon, which controls access to Internet services
<i>/etc/inittab</i>	Configuration for the <code>init</code> daemon, which controls executing processes
<i>/etc/issue</i>	Linux kernel and distribution version
<i>/etc/lilo.conf</i>	Loader (<i>lilo</i>) configuration file

File	Description
<i>/etc/login.defs</i>	Options for <code>useradd</code> and related commands
<i>/etc/minicom.users</i>	Userids allowed to use <code>minicom</code>
<i>/etc/mtab</i>	Mounted filesystems
<i>/etc/passwd</i>	User account information
<i>/etc/printcap</i>	Printer options and capabilities
<i>/etc/profile</i>	Default environment for users of BASH shell
<i>/etc/rc*.d</i>	Scripts for system and process startup and shutdown
<i>/etc/rc.boot</i>	Scripts for system boot
<i>/etc/shadow</i>	Secure user account information
<i>/etc/skel</i>	Skeleton files used to establish new user accounts

File	Description
<i>/etc/terminfo</i>	Terminal capabilities and options
<i>/etc/X11/XF86Config</i>	X configuration file
<i>/var/log/apache/access.log</i>	Log of web server access
<i>/var/log/apache/error.log</i>	Log of web server errors
<i>/var/log/messages</i>	System log
<i>/var/spool/cron</i>	Directory for <code>at</code> and <code>cron</code> configuration files



Learning Debian GNU/Linux

By [Bill McCarty](#)

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

C. The Debian Package Management Utilities

Contents:

[Packages](#)

[The Package Management Tools](#)

[Using `dpkg`](#)

[Using `dselect`](#)

[Using `apt-get`](#)

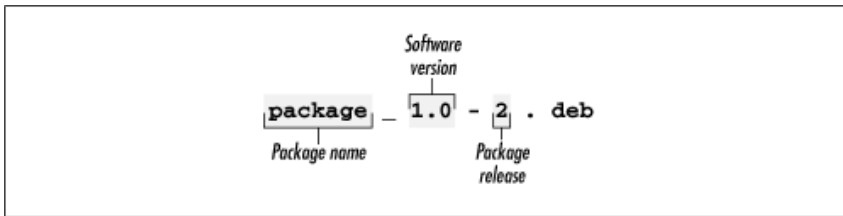
[Package Contents and Installation Commands](#)

This appendix introduces you to Debian's package management facilities, which help you manage software applications. Suppose, for example, that after installing Linux, you discover you need an application that you omitted; you can find the missing application's package and use the package management facilities to quickly and easily install the application. Similarly, when a new version of an application becomes available, the package management facilities helps you upgrade painlessly, by preserving the application's configuration files. The package management facilities also let you query the status of your system, helping you determine whether important files have been deleted.

C.1 Packages

A Debian package (or more simply, a package or a deb) is a file that contains files necessary to install an application or software unit. Debian packages are generally named using a convention that lets you determine the name of the package, the version of the software, and the release number of the package. [Figure C.1](#) shows how the components of a package name are arranged.

Figure C.1: The structure of a package name



The important virtue of packages is that they contain meta-information; that is, information about their contents. For example, each package contains a list of other packages needed for correct operation. Similarly, each package contains a list of other packages that conflict with its operation. This meta-information greatly simplifies system administration, which otherwise can become quite a challenge. Each package also contains a checksum, that helps protect package users against viruses and other sorts of tampering with package contents.

Debian packages have several important advantages relative to those used by other package management schemes. To learn more about these, see the comparison of package formats authored by Joey Hess, at <http://kitenet.net/~joey/pkg-comp/>.

C.2 The Package Management Tools

Debian GNU/Linux provides three main package management tools:

`dpkg`

The original package management tool, which is used from the command line. The newer package management tools, which most users find more convenient, invoke `dpkg` on behalf of the user. However, `dpkg` provides access to some functions - particularly query functions - not available using the more user-friendly tools.

`dselect`

A graphical front-end to `dpkg`, which lets the user browse a list of available packages and select packages for installation or removal.

`apt-get`

A command-line tool that lets the user maintain a list of sources from which to obtain packages. Sources can be files on mounted filesystems, files on CD-ROM, files on FTP servers, and so on. When the user requests that a package be installed, `apt-get` locates the package file on one of its servers, downloads the package if necessary, and installs the package.

This appendix does not attempt to cover all the features and options of each tool. Instead, the material is organized around a set of common system administration tasks, usually presenting only one technique for dealing with each task. As you learn more about Debian, you'll discover other ways of performing these tasks.

Since package management generally involves manipulation of files with restricted permissions, most package management operations must be performed by `root`.

C.3 Using dpkg

You might think that `dpkg` has been superseded by the more recent package management tools, `dselect` and `apt-get`, which stress ease of use. However, `dpkg` remains a good choice for performing several common package management tasks.

C.3.1 Installing a Package

If you have a package file containing a package you want to install, the simplest way to install the package is to use the `dpkg` command:

```
dpkg --install  
packagefile
```

where *packagefile* stands for the name of the package file, which generally ends with the characters *.deb*. If all the prerequisite packages have already been installed and if the package does not conflict with any installed packages, the command will unpack the package files, move them to their proper locations, and execute the scripts necessary to configure the package.

If your system lacks a prerequisite package or if the specified package conflicts with a package installed on your system, `dpkg` will report the error and terminate. If the problem is the lack of one or more prerequisite packages, you can obtain and install them, and then install the desired package. If the problem is a package conflict, you must decide which of the conflicting packages you want. If you decide to remove an installed package, you can do so using the technique described in the following subsection.

C.3.2 Removing a Package

To remove an installed package, use the command

```
dpkg --remove  
package
```

This command does not remove package configuration files, which may facilitate subsequent re-installation of the package. If you want to remove the configuration files as well, use the command:

```
dpkg --purge  
package
```

C.3.3 Querying the Package Database

The Debian package management facility maintains a database that contains information about installed packages. You can use the `dpkg` command to query this database.

C.3.3.1 Printing the description of a package

To print the description of a package, issue the following command:

```
dpkg --print-avail  
package
```

where *package* specifies the name of the package. For example, to print the description of the package *gnome-guile*, issue the command:

```
dpkg --print-avail gnome-guile
```

C.3.3.2 List packages by name

To list known packages by name, issue the following command:

```
dpkg -l  
pattern
```

where *pattern* is a single-quoted string that specifies a pattern. Only packages with names matching the pattern will be listed. The pattern can include wildcard characters such as an asterisk (*), which substitutes for any string of characters. For example, the pattern `'apache*'` matches package names beginning with *apache*.

The listing presents the following information:

Selection status

Indicates the selection status established using `dselect`, which may be any one of:

Unknown

Indicates that the selection status is not known.

Install

Indicates that the package is marked for installation.

Remove

Indicates that the package is marked for removal.

Purge

Indicates that the package and its configuration files are marked for removal.

Status

Indicates the installation status of the package, which may be any one of:

Not installed

Indicates that the package has not been installed.

Installed

Indicates that the package has been successfully installed.

Config-files

Indicates that only the package's configuration files are currently installed.

Unpacked

Indicates that the package has been unpacked, in preparation for installation.

Failed-config

Indicates that the package has been installed, but its configuration script failed.

Half-installed

Indicates that an attempt to install the package failed.

Error

Indicates the error status of the package, which may be one or more of:

None

Indicates no error is associated with the package.

Hold

Indicates that the package has been placed on hold, so that it can be neither installed nor removed.

Reinstallation required

Indicates that the package must be re-installed.

Name

Gives the name of the package.

Version

Gives the version number of the package.

Description

Gives a brief description of the package. Descriptions are generally available only for installed packages.

If the command produces too much output to conveniently view, pipe its result through the `more` command, which lets you page through the output:

```
dpkg -l  
pattern | more
```

If you want to view only installed packages, issue a command such as:

```
dpkg -l  
pattern | grep '^i' | more
```

For example, to view installed packages with names beginning with *gnome*, issue the following command:

```
dpkg -l 'gnome*' | grep '^i' | more
```

The pattern `'*'` matches any package name, so the following command prints information about every installed package:

```
dpkg -l '*' | grep '^i' | more
```

C.3.3.3 Report status of a package

To report the status of a package, issue the following command:

```
dpkg --status  
package
```

where *package* specifies the name of the package.

For example, to report the status of the *gnome-guile* package, issue the command:

```
dpkg --status gnome-guile
```

C.3.3.4 List files installed from a package

To list the files installed from a specified package, issue the command:

```
dpkg --listfiles  
package
```

where *package* specifies the name of the package.

For example, to list the files installed from the *gnome-guile* package, issue the command:

```
dpkg --listfiles gnome-guile
```

C.4 Using dselect

The `dselect` tool provides an easy-to-use, character-based graphical front-end for accessing `dpkg`. To launch `dselect`, issue the command:

```
dselect
```

[Figure C.2](#) shows the screen that appears. The screen presents a simple menu having six items:

Access

This menu item lets you choose the method used to access package files.

Update

This menu item lets you update the list of available packages.

Select

This menu item lets you choose packages for installation or removal.

Install

This menu item initiates installation of selected packages.

Config

This menu item initiates configuration of installed packages.

Remove

This menu item initiates removal of packages selection for removal.

Quit

This menu item exits `dselect`.

The menu items are generally used in the order in which they are presented.

Figure C.2: The dselect screen

```
Debian Linux 'dselect' package handling frontend.

# 0. (0)Access Choose the access method to use.
1. (U)Update Update list of available packages, if possible.
2. (I)Select Request which packages you want on your system.
3. (I)Install Install and upgrade wanted packages.
4. (C)onfig Configure any packages that are unconfigured.
5. (R)emove Remove unwanted software.
6. (Q)uit Quit dselect.

Use ^P and ^N, cursor keys, initial letters, or digits to select;
Press ENTER to confirm selection. ^L to redraw screen.

Version 1.4.8.34 (i386 elf). Copyright (C) 1994-1996 Ian Jackson. This is
free software; see the GNU General Public Licence version 2 or later for
copying conditions. There is NO warranty. See dselect --licence for details.
```

C.4.1 Choosing the Access Method

To choose the access method, use the arrow keys to highlight the Access menu item and press **Enter**. The screen shown in [Figure C.3](#) appears.

Figure C.3: Choosing the access method

```
dselect - list of access methods
Abbrev. Description
cdrom Install from a CD-ROM.
multi_cd Install from a CD-ROM set.
nfs Install from an NFS server (not yet mounted).
multi_nfs Install from an NFS server (using the CD-ROM set) (not yet moun
harddisk Install from a hard disk partition (not yet mounted).
mounted Install from a filesystem which is already mounted.
multi_mount Install from a mounted partition with changing contents.
floppy Install from a pile of floppy disks.
ftp Install using ftp.
# apt APT Acquisition (file,http,ftp)

Access method 'apt'.
apt - APT Acquisition (file,http,ftp)

The APT installation method encompasses most other installation methods
under the umbrella of the new Package Acquisition code. This method allows
installation from locations in the filesystem, ftp and http URLs, supports
full installation ordering and dependency checking as well as multiple
sources. See the man pages apt-get(8) and sources.list(5)

HTTP proxies can be used by setting http_proxy="http://proxy:port/" before
running DSelect. FTP proxies require special configuration detailed in the
apt.conf(5) man page (see /usr/doc/apt/examples/apt.conf)

Explanation of apt
```

The most flexible access method - and the method that's generally recommended - is *apt*. Other available options include:

cdrom

Lets you install packages from a CD-ROM. This access method has been deprecated; you should use *multi_cd* instead.

multi_cd

Lets you install packages from a multi-volume set of CD-ROMs.

nfs

Lets you install packages residing on an NFS server. This access method has been deprecated; you should use *multi_nfs* instead.

multi_nfs

Lets you install packages residing on an NFS server that has access to a multi-volume set of packages.

harddisk

Lets you install packages residing on a hard disk partition that is not currently mounted. This access method has been deprecated; you should use *apt* or *multi_mount* instead.

mounted

Lets you install packages residing on a currently mounted filesystem. This access method has been deprecated; you should use *apt* or *multi_mount* instead.

multi_mount

Lets you install packages from a multi-volume set, one volume of which is currently mounted.

floppy

Lets you install packages from a set of floppy diskettes.

ftp

Lets you install packages residing on an FTP server.

To choose an access method, use the arrow keys to highlight the appropriate menu item and press **Enter**.

If you selected the *apt* access method, you'll be asked if you want to change the *sources.list* file. If you've previously configured the file, you should respond *No*. If you've not configured the file, you can respond *Yes*, which initiates a dialog that builds a simple configuration. Here's a sample dialog that shows the responses you should give to install packages from the CD-ROM diskette that accompanies this book:

```
I see you already have a source list.
```

```
-----
```

```
source list displayed here: contents vary
```

```
-----
```

```
Do you wish to change it?[y/N]
```

y

```
Set up a list of distribution source locations
```


Please give the base URL of the debian distribution.
The access schemes I know about are: http ftp file

For example:

```
file:/mnt/debian,  
ftp://ftp.debian.org/debian,  
http://ftp.de.debian.org/debian,
```

URL [http://http.us.debian.org/debian]:

file:/cdrom

Please give the distribution tag to get or a path to the
package file ending in a /. The distribution
tags are typically something like: stable unstable frozen non-US

Distribution [stable]:

stable

Please give the components to get
The components are typically something like: main contrib non-free

Components [main contrib non-free]:

main contrib

Would you like to add another source?[y/N]

N

The sample dialog assumes that your CD-ROM has been mounted as */cdrom*. If your CD-ROM is mounted differently, you'll need to revise the dialog.

After `dselect` records your choice of access method, the main menu screen reappears.

C.4.2 Updating Information on Available Packages

After selecting the access method, you should instruct `dselect` to update information on available packages. To do so, use the arrow keys to highlight the Update menu item and press **Enter**. After a short time, the main menu will reappear.

C.4.3 Choosing Packages for Installation or Removal

Once you've updated the information on available packages, you're ready to select packages for installation or removal. To do so, use the arrow keys to highlight the Select menu item and press **Enter**. The screen shown in [Figure C.4](#) appears.

Figure C.4: The introduction screen

```
Help: Introduction to package list
Welcome to the main package listing. Please read the help that is available !

You will be presented with a list of packages which are installed or available
for installation. You can navigate around the list using the cursor keys,
mark packages for installation (using '+') or deinstallation (using '-').

Packages can be marked either singly or in groups; initially you will see that
the line 'All packages' is selected. '+', '-' and so on will affect all the
packages described by the highlighted line. Use 'o' to change the order of the
list (this also changes which kinds of group selections are possible).

(Mainly for new installations:) Standard packages will be requested by default.
Use capital 'D' or 'R' key to override this - see the keybindings help screen.

Some of your choices will cause conflicts or dependency problems; you will be
given a sub-list of the relevant packages, so that you can solve the problems.

When you are satisfied with your choices you should press Return to confirm
your changes and leave the package listing. A final check on conflicts and
dependencies will be done - here too you may see a sublist.

Press Space to leave help and enter the list; press '?' at any time for help.
? - help menu  Space - exit help  . - next help  or a help page key
```

This screen provides an overview of the package selection screens. When you've read its contents, press **Space** to go to the package selection screen, which is shown in [Figure C.5](#).

Figure C.5: The package selection screen

```
dselect - main package listing (by availability) mark: / - / verbase: o help: ?
----- Up to date installed packages -----
*** Req base aalib1 1.2-8 1.2-8 ascii art library
*** Req base adduser 3.8 3.8 Add users and groups to t
*** Req base ac 962-21.1 962-21.1 Anthony's Editor -- a tin
*** Opt uch apache 1.3.6-15 1.3.3-7 Versatile, high-performan
*** Opt uch apache-canna 1.3.6-15 1.3.3-7 Support files for all Apa
*** Opt admin apt 0.3.11 0.1.9 Front-End for dpkg
*** Opt x11 asclock 1.0R10-1 1.0R10-1 A clock designed with the
*** Imp admin at 3.1.8-4 3.1.8-4 Delayed job execution and
*** Req base base-files 2.01.0 2.01.0 Debian Base System Miscel
*** Req base base-passwd 2.01.3.3 2.01.3.3 Debian Base System Passwa
*** Req base bash 2.01.1-4.1 2.01.1-4.1 The GNU Bourne Again SHel
*** Imp math bc 1.05a-3 1.05a-3 An arbitrary precision ca
*** Std devel bin86 0.14.3-1 0.14.3-1 16-bit assembler and load
*** Std devel binutils 2.9.1.0.19a 2.9.1.0.19a The GNU assembler, linker
*** Std devel bison 1.25-13 1.25-13 A parser generator that i
base-files - Debian Base System Miscellaneous Files
This package contains the Debian Base System Miscellaneous Files, such as
/etc/debian_version, /etc/issue, /usr/doc/copyright/*, etc.
Description of base-files
```

To use the package selection screen, use the arrow keys to highlight a package in the upper part of the screen. The lower part of the screen will display information about the highlighted package. To select the package for installation, press **+**; to select an installed package for removal, press **-**.

You can search the package database by typing a slash (/) followed by the string for which you wish to search. To find successive instances of the same string, type a backslash (\). For example, to find the first package that contains the string *gnome* in its name or description, type **/gnome** and press **Enter**.

If you select for installation a package that requires one or more other packages that are not installed, a dependency conflict results. Similarly, a dependency conflict results if you mark for removal a package required by an installed package or if you mark for installation a package that conflicts with an installed package.

When `dselect` detects a dependency conflict, it presents the screen shown in [Figure C.6](#).

Figure C.6: The dependency help screen

```
Help: Introduction to conflict/dependency resolution sub-list
Dependency/conflict resolution - Introduction.

One or more of your choices have raised a conflict or dependency problem -
some packages should only be installed in conjunction with certain others, and
some combinations of packages may not be installed together.

You will see a sub-list containing the packages involved. The bottom half of
the display shows relevant conflicts and dependencies; use 'i' to cycle between
that, the package descriptions and the internal control information.

A set of 'suggested' packages has been calculated, and the initial markings in
this sub-list have been set to match those, so you can just hit Return to
accept the suggestions if you wish. You may abort the change(s) which caused
the problem(s), and go back to the main list, by pressing capital 'X'.

You can also move around the list and change the markings so that they are more
like what you want, and you can 'reject' my suggestions by using the capital
'D' or 'R' keys (see the keybindings help screen). You can use capital 'Q' to
force me to accept the situation currently displayed, in case you want to
override a recommendation or think that the program is mistaken.

Press Space to leave help and enter the sub-list; remember: press '?' for help.
/? = help menu      Space = exit help      . = next help      or a help page key
```

The same screen appears if you select for installation a package that specifies recommended or suggested packages to be installed with it. A recommended package is one that most users install with the recommending package. A suggested package is one that is related to the suggesting package; suggested packages often extend or complement the functionality of the suggesting package.

When you press **Space**, you're presented with the conflict resolution screen, shown in [Figure C.7](#). This screen lets you quickly select packages for installation or removal the packages involved in a dependency conflict. The screen also presents default choices for recommended and suggested packages.

Figure C.7: The dependency resolution screen

```
dselect - recursive package listing                                mark!+/-/- verbose? help:?
SIOM PFI Section Package Description
--
Std devel gcc The GNU (G)cc C compiler.
Std devel g++ The GNU (g++) C++ compiler.
Std devel gcc The GNU C compiler.
Std devel libstdc++2.9 The GNU stdc++ library (development files)
Std utilz dpkg-dev Package building tools for Debian Linux
Std devel flex A fast lexical analyzer generator.
Std devel libc6-dev GNU C library: Development libraries and header f
Opt devel debsake Debianizing Tool and automated binary generation
Opt admin apt Front-End for dpkg
Opt x11 xlib6g-dev include files and libraries for X client developm
Std x11 xfree86-comm X Window System (XFree86) infrastructure
Opt x11 xmanpages manual pages for X developers
Opt x11 xdm X display manager

egcc not installed: purge (was: purge). Extra
egcc depends on g++ (>= 2.91.60)
egcc suggests egcs-docs (>= 2.91.60)
gcc conflicts with egcc << 2.91.63-1.1>
egcc recommends libc-dev

Interrelationships affecting egcc
```

Using the arrow keys to highlight an entry lets you view a list of dependencies related to the entry. By pressing **+** or **-** you can select packages for installation or removal, just as on the selection screen. When you're done working out dependencies, you can press **Space** to return to the selection screen.

C.4.4 Exiting the Select Function

You can select the Select function in any of several ways. Pressing **Space** returns you to the main menu, where you can initiate installation or removal of packages. Pressing **x** cancels your selections and returns you to the main menu. This feature is useful if you change your mind about installing a package, possibly owing to conflicts associated with the package.

C.4.5 Installing Packages

To begin installing the selected packages, use the arrow keys to highlight the Install menu item and press **Enter**. As packages are downloaded or installed, you'll see messages on the screen.

If you're using the *apt* access method, selecting Install actually initiates installation, removal, and configuration of packages. You can exit `dselect` after the installation process completes.

If you're using some other access method, `dselect` may not install every selected package in a single operation. When the installation process completes, you should select Install and see if more packages are installed. When you select Install and no more packages are installed, you can proceed to the subsequent steps: configuration and removal.

When the installation process is complete, `dselect` prompts you to press **Enter** to return to the main menu.

C.4.6 Configuring Packages

To begin configuring the installed packages, use the arrow keys to highlight the Configure menu item and press **Enter**. Packages that require configuration will prompt you for configuration choices. When the configuration process is complete, `dselect` prompts you to press **Enter** to return to the main menu.

C.4.7 Removing Packages

To begin removing the packages selected for removal, use the arrow keys to highlight the Remove menu item and press **Enter**. When the removal process is complete, `dselect` prompts you to press **Enter** to return to the main menu.

C.4.8 Exiting dselect

To exit `dselect`, use the arrow keys to highlight the Quit menu item and press **Enter**.

C.5 Using apt-get

The `dselect` program is useful, because it lets you browse a list of available packages, viewing their descriptions and dependencies, and selecting desired packages for installation. However, if you know the name of a package you want to install, `apt-get` is often the easiest way to install it. Before using `apt-get`, you must configure the *sources.list* file. This same file is used when you choose the apt access method of `dselect`. Even if you don't plan on using `apt-get`, you'll find the information in the following subsection useful.

C.5.1 Configuring the *sources.list* File

The *sources.list* file resides in the `/etc/apt` directory. Like most other Linux configuration files, it can be revised by using an ordinary text editor, such as `ae`.

The file contains a series of lines, each specifying a source for packages. The lines are consulted serially, so it's usually advantageous to place lines that specify local sources - such as a CD-ROM - ahead of lines that specify remote sources. Doing so can save many minutes of download time.

Each line has the form:

```
deb
uri distribution components
```

The *uri* is a universal resource identifier (URI) that specifies the computer on which the packages reside, the location of the packages, and the protocol used for accessing the packages. It has the following form:

```
protocol://
host/
path
```

Four protocols - sometimes called URI types - are recognized:

cdrom

A local CD-ROM drive.

file

A directory of the local filesystem.

http

A Web server.

ftp

An FTP server.

The *host* part of the URI and the preceding pair of slashes (//) are used only for the *http* and *ftp* protocols. There, the *host* part of the URI gives the name of the host that contains the packages.

The *path* part of the URI always appears, with the preceding slash (/). It specifies the absolute path of the directory that contains the packages.

Here are some examples of typical URIs:

```
cdrom:/cdrom
cdrom:/mnt/cdrom
file:/mnt
file:/debian
http://www.us.debian.org/debian
http://non-us.debian.org/debian-non-US
ftp://ftp.debian.org/debian
ftp://nonus.debian.org/debian-non-US
```

The distribution part of a *sources.list* line specifies the distribution release that contains the packages. Typical values include:

stable

The latest stable release; that is, one that is commonly regarded as having sufficiently few serious bugs for everyday use.

unstable

The latest unstable release. This release sometimes contains serious bugs and should not be installed by users who require high levels of system availability or reliability.

The components part of a *sources.list* line specifies the parts of the distribution that will be accessed. Typical values include:

main

The main set of packages.

contrib

Packages not an integral part of the distribution, but which may be useful.

non-free

Packages that contain software distributed under terms too restrictive to allow inclusion in the distribution, but which may be useful.

A typical *sources.list* file might contain the following entries:

```
deb file:/cdrom stable main contrib
deb http://www.us.debian.org/debian stable main contrib non-free
deb http://non-us.debian.org/debian-non-US stable non-US
```

This configuration allows rapid access to the distribution packages contained on the local CD-ROM. It also allows convenient access via the network to other packages and more recent package versions stored on web servers.

C.5.2 Using apt-get

Once you've configured *sources.list*, you can use `apt-get` to update information on available packages, to install a package, or to upgrade installed packages.

C.5.2.1 Updating Information on Available Packages

To update information on available packages, issue the following command:

```
apt-get update
```

C.5.2.2 Installing a Package

To install a specified package, issue the following command:

```
apt-get install
package
```

where *package* specifies the name of the package to be installed.

C.5.2.3 Upgrading Installed Packages

To automatically upgrade all installed packages to the latest available version, issue the following command:

```
apt-get upgrade
```


C.6 Package Contents and Installation Commands

This section gives details on the installation and configuration of packages mentioned throughout the book.

C.6.1 Chapter 5: Installing and Configuring X Windows

[Table C.1](#) describes the X servers included in Debian GNU/Linux and identifies the supported cards and chipsets.

Table C.1: X Servers for Accelerated Graphics Cards

Package	Supported Cards and Chipsets
<i>xserver-8514</i>	IBM 8514/A and other compatible cards
<i>xserver-AGX</i>	All AGX cards
<i>xserver-II28</i>	#9 Imagine 128 (including Series II) cards
<i>xserver-Mach32</i>	ATI cards using the Mach32 chipset
<i>xserver-Mach64</i>	ATI cards using the Mach64 chipset
<i>xserver-Mach8</i>	ATI cards using the Mach8 chipset
<i>xserver-P9000</i>	Diamond Viper and other P9000 cards (excluding cards using the 9100)
<i>xserver-S3</i>	#9 cards, most Diamond cards, some Orchid cards, and others

Package	Supported Cards and Chipsets
<i>xserver-S3V</i>	Cards using the S3 ViRGE chipset, including the DX, GX, and VX
<i>xserver-W32</i>	ET4000/W32 cards, excluding standard ET4000 cards

[Table C.2](#) describes other X-related packages you should install in order to run X.

Table C.2: Recommended X-Related Packages

Package	Description
<i>xbase-clients</i>	Miscellaneous X clients
<i>xcontrib</i>	XFree-contributed clients
<i>xfonts-100dpi</i>	100-dpi fonts for X servers
<i>xfonts-75dpi</i>	75-dpi fonts for X servers
<i>xfonts-base</i>	Standard fonts for X servers
<i>xfonts-scalable</i>	Scalable fonts for X servers
<i>xfree86-common</i>	Basic components of the X Windows System
<i>xlib6g</i>	Shared libraries for X clients

Package	Description
<i>xpm4g</i>	X Pixmap libraries
<i>xserver-common</i>	Components common to X servers
<i>xterm</i>	X terminal emulator

C.6.2 Chapter 6: Using X Windows

[Table C.3](#) describes the packages you should install in order to run GNOME.

Table C.3: GNOME-Related Packages

Package	Description
<i>gmc</i>	GNOME file manager
<i>task-gnome-apps</i>	GNOME applications and utilities
<i>task-gnome-desk</i>	GNOME desktop
<i>task-gnome-game</i>	GNOME games
<i>task-gnome-wm</i>	GNOME window managers
<i>wmaker-gnome</i>	Window Maker compiled with GNOME support
<i>wmaker-usersguide-ps</i>	Window Maker Users' Guide

C.6.3 Chapter 8: Using Linux Applications and Clients

Debian's rigid adherence to exclusively open-source software means that many programs, like WordPerfect or Applixware, are not (currently) available as Debian packages. Often, though, a pre-compiled binary is available; follow the instructions (usually found in a README file) to install the software.

C.6.3.1 Installing Star Office

To install StarOffice, install the Debian package `staroffice3` and follow the instructions that accompany it. You'll also need to download the StarOffice files from Sun's web site, <http://www.sun.com/products/staroffice/>.

C.6.3.2 Installing Applix

To install Applixware, download the files from the Applix web site, <http://www.applix.com/>.

C.6.3.3 Installing WordPerfect for Linux

To install WordPerfect for Linux, download the files from Corel's Web site, <http://www.corel.com/>.

C.6.3.4 Installing the Mesa and WINE packages

[Table C.4](#) lists the packages you should install to experiment with WINE. Mesa and WINE are frequently updated. Ideally, you should download the latest versions of Mesa and WINE from <http://www.winehq.com/>. However, the WINE developers do not currently make Debian packages available.

Table C.4: WINE-Related
Packages

Package	Description
<i>mesa3g</i>	Mesa graphics library
<i>wine</i>	WINE

Package	Description
<i>wine_doc</i>	WINE documentation

C.6.4 Chapter 9: Playing Linux Games

DOOM and Quake II pose different challenges while installing. The following sections will lead you through the installation for each.

C.6.4.1 Installing Doom

You can download Linux Doom and the required IWAD file from id Software's web site, <http://www.idsoftware.com/archives/>. You can download either an SVGA (*linuxsdoom*) or X11 (*linuxxdoom*) version of the game. The X11 version requires some deprecated libraries that are no longer part of the Debian distribution; it can't be run without extraordinary measures. However, id Software has released the source code for Doom, and at least one Debian developer has compiled an X11 version of Doom that's compatible with the current libraries. See the mail list archive on the Debian web site, <http://www.debian.org/>.

To install the files, move to the directory into which you downloaded them and issue the following commands:

```
tar zxvf linuxsdoom.tar.gz
gunzip doom-1.8.wad.gz
```

If you prefer, you can download and install the DOSEMU package, which lets you run the MS-DOS version of Doom under Linux.

C.6.4.2 Installing the quake2 package

To install Quake II under Linux, you'll need the `quake2` package and the game data. You can obtain the game data in any of several ways. The best way is by purchasing the retail Quake II CD-ROM. However, if you want merely to try out the game, you can download the Windows 9x demo file.

To install Quake II, issue the following command:

```
apt-get quake2
```

If your video card has Voodoo acceleration and you want to enable the acceleration, you'll face several challenges. See `/usr/doc/quake2/README` for details. Also see the Linux Quake HOWTO (<http://www.linuxquake.com/howto/Quake-HOWTO.html>).

C.6.4.3 Installing game data from the Quake II CD-ROM

If you have the Quake II CD-ROM, mount it and copy the files in the *Install/Data* directory (and all its subdirectories) to the current working directory. To copy the subdirectories, use the `-r` argument:

```
mount -t iso9660 /dev/cdrom /cdrom
cp -r /cdrom/Install/Data/* .
```

Now you can delete some unneeded files:

```
rm -f /usr/local/quake2/*.dll
rm -f /usr/local/quake2/quake2.exe
rm -f /usr/local/quake2/baseq2/gamex386.dll
```

C.6.4.4 Installing Game Data from the Quake II Demo

If you have the self-extracting demo file (`q2-314-demo-x86.exe` or a similarly named file) rather than the CD-ROM, place the file in the current working directory and use the `unzip` utility to extract its contents:

```
unzip q2-314-demo-x86.exe
```

If your file has a different name, adjust the command accordingly. If your system doesn't include the `unzip` package, you'll have to install it before you can extract the contents of the demo file.

Then, rearrange some files and delete others:

```
rm -rf Splash Setup.exe
mv Install/Data/baseq2 .
mv Install/Data/DOCS docs
rm -rf Install
rm -f baseq2/gamex86.dll
```

C.6.5 Chapter 10: Setting Up a Linux-Based LAN

These packages will help establish a local network.

C.6.5.1 Installing the lynx package

When installing the `lynx` package, the configuration script asks what default URL should be used when invoking `lynx`. If you have a favorite web page that's available at all times - even when you're not dialed into your ISP, for example - you can specify that page. Otherwise, you may find it convenient to specify the URL file:/usr/doc/lynx/lynx_help/lynx_help_main.html, the main `lynx` help page.

C.6.5.2 Installing the samba package

To install `samba`, install the packages listed in [Table C.5](#).

Table C.5: Samba-Related Packages

Package	Description
samba	A Microsoft-compatible network server.
samba-doc	Samba document.
smbclient	Samba client.
swat	Web-based configuration tool for Samba.

The `samba` configuration script will ask if you want to run `samba` as a daemon or from `inetd.conf`. If your expected usage is quite low, it's appropriate to run `samba` from `inetd.conf`; otherwise you should run it as a daemon.

The `samba` configuration script will also ask if you are going to use encrypted passwords. You should respond yes in order to configure `samba` to work properly with Windows 98 and Windows NT. To establish an encrypted password for a user, issue the `smbpasswd` command. For example:

```
debian:/etc/samba# smbpasswd
debian:/etc/samba# smbpasswd bmccarty
New SMB password:
Retype new SMB password:
Password changed for user bmccarty.
```

To use `swat`, you must add a line to the `/etc/services` file:

```
swat                901/tcp
```

Place the new line so that the number of the related service (901) is properly sequenced.

You must also add a line to */etc/inetd.conf*:

```
swat      stream  tcp      nowait.400      root  /usr/sbin/swat swat
```

You may find it convenient to add the line near the line that refers to `smbd`, if any.

At present, `swat` is considered unstable, though the author has used it extensively without difficulty. Nevertheless, you may want to obtain an updated version - when available - from the Debian web site, <http://www.debian.org/>.

C.6.6 Chapter 11: Getting Connected to the Internet

While most of these packages are available as Debian packages, Netscape Navigator is not (at least not yet). More information is available in each subsection.

C.6.6.1 Installing the `wvdial` package

To install `wvdial`, issue the following command:

```
apt-get install wvdial
```

The installation script asks you for:

- Your Internet service provider's dial-in phone number
- Your login name
- Your password

Next, the install script scans your serial ports to locate your modem and establish its configuration. Generally, the scan is successful; however, some systems hang, forcing you to reboot the system and configure the modem manually.

Here's a typical *wvdial.conf* file, as built by the configuration script:

```
[Dialer Defaults]
Phone = 1-714-555-0612
Username = bmccarty
Password = abrupti
New PPPD = yes
Modem = /dev/ttyS0
Baud = 57600
```



```
Init1 = ATZ
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 S11=55 +FCLASS=0
```

C.6.6.2 Installing Netscape Navigator

Netscape Navigator is subject to distribution restrictions, which prevent it from being included in the Debian GNU/Linux distribution. However, you can install Navigator by downloading and installing the packages *netscape-base-4*, *netscape-base-45*, *netscape-java-45*, *navigator-base-45*, *navigator-nethelp45*, and *navigator-smotif45*, which are available on the Debian web site, <http://www.debian.org/>.

C.6.6.3 Installing the seyon package

To install *seyon*, simply install the package by using the command:

```
apt-get --install seyon
```

A sample configuration dialog follows:

```
Do you have a color X11 system? (Y/n)y
Where is your modem located?
(0) /dev/ttyS0 (equivalent to DOS COM1)
(1) /dev/ttyS1 (equivalent to DOS COM2)
(2) /dev/ttyS2 (equivalent to DOS COM3)
(3) /dev/ttyS3 (equivalent to DOS COM4)
Enter (0/1/2/3) -> 1
Modem configured for: /dev/ttyS1

*** Important ***
Users must be added to the 'dialout' group to allow access to the modem
device.
```

To add the user *bmccarty*, for example, to the *dialout* group, issue the command

```
addgroup bmccarty dialout
```

C.6.6.4 Installing the minicom package

To install *minicom*, issue the following commands:

```
apt-get install minicom
```

The *minicom* configuration will ask if you want to use the **Meta (Alt)** key as the *minicom* command key. Doing so precludes you from using *minicom* under X, so you should normally respond *No*. After installing *minicom* you can set system-wide options by issuing the command

```
minicom -s
```

You must be logged in as `root` to accomplish this. As with `seyon`, ordinary users cannot access `minicom` unless they are members of the `dialout` group.

C.6.7 Chapter 12: Setting Up a Linux-Based WAN

U.S. law forbids the export of strong cryptography software. Consequently, the secure shell packages cannot be included on the CD-ROM accompanying this book.

C.6.7.1 Installing the apache package

To install the Apache web server, issue the following commands:

```
apt-get install apache
```

The configuration script will ask:

The email address of the server administrator (ServerAdmin).

Specify your email address.

The absolute path of the directory that will be the root directory (DocumentRoot).

Specify `/var/www` or another directory.

What port Apache should listen on

Specify 80 or another port of your choice. If you specify a port other than 80, users must include the port in the URL of each page accessed.

Whether you want to manually choose the modules to be loaded

Normally, you should respond No.

Whether to save changes to the configuration files

Normally, you should respond Yes.

Whether to restart Apache

Normally, you should respond Yes.

C.6.7.2 Installing the ssh client and server package

Owing to U.S. export laws, these packages do not appear on the CD-ROM. To install the Secure Shell client and server, issue the following commands:

```
apt-get install ssh
```

Your *sources.list* file must point to a source for non-US packages.



Learning Debian GNU/Linux

By Bill McCarty

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

D. Managing the Boot Process

Contents:

[Booting Linux](#)

[Boot Disks](#)

[The lilo Loader](#)

[The loadlin Loader](#)

[Boot Parameters](#)

[Using Loadable Ethernet Drivers](#)

In this appendix, you'll learn more about how to boot a Linux system; in particular, you'll learn more about configuring your computer system to boot any of several operating systems. The chapter focuses on `lilo` and `loadlin`, the most popular utilities for booting Linux systems, explaining their capabilities and options in considerable detail.

D.1 Booting Linux

When you boot a PC, you cause it to execute a small program known as a *boot loader*. The purpose of the boot loader is to locate and read into memory the first stage of an operating system and transfer control to it. The operating system then locates and reads its remaining components as needed.

The simplest way to boot Linux is by using a floppy diskette. By doing so, you're able to leave the boot information on your hard drive untouched, ensuring that you can still boot Microsoft Windows or another operating system stored on the same hard drive. Moreover, some operating systems and virus protection programs prevent modification of the boot information on your hard drive. By booting from a floppy diskette, you avoid several potential problems.

However, many users find booting from a floppy diskette slow or inconvenient. You don't have to boot Linux from a floppy diskette; you can boot Linux in any of several other ways. The two most popular alternatives are by using `lilo`, which replaces the boot loader stored on your hard drive, or `loadlin`, which lets you first boot DOS and then boot Linux from DOS.

This chapter cannot describe the entire range of issues involved in booting Linux. Much of the information in this chapter is taken from several Linux HOWTOs that contain additional useful information on booting Linux:

- [BootPrompt-HOWTO](#)
- [CD-Writing-HOWTO](#)
- [CDROM-HOWTO](#)
- [Ethernet-HOWTO](#)
- [Ftape-HOWTO](#)

- [Hardware-HOWTO](#)
- [Multi-Disk-HOWTO](#)
- [PCI-HOWTO](#)
- [PCMCIA-HOWTO](#)

D.2 Boot Disks

Even if you don't want to boot Linux from a floppy diskette, you should create and keep on hand a Linux boot floppy. If something goes wrong with your system, preventing you from booting in the normal way, you may be able to boot your system by using the floppy diskette. Then, you can diagnose and repair the problem and get back to business as usual.

D.2.1 Creating a Boot Disk

The Debian GNU/Linux install program gives you the option of creating a boot diskette when you install Linux. You should exercise this option each time you install Linux, so that you have a fresh boot disk containing software consistent with that stored on your hard drive.

However, you can easily create a boot diskette after the installation is complete. To do so, insert a blank floppy diskette into your system's floppy drive. Log on as `root` and issue the following command:

```
mkboot  
kernal
```

For `kernal`, substitute the name of the file that contains the kernel you want to place on the boot disk, usually `vmlinuz`.

D.2.2 Using a Boot Disk

Insert the boot disk into your system's floppy drive. If your system is turned off, power up your system. If your system is turned on, first shut down the active operating system in the proper manner, then restart the system. Linux should then boot from the floppy diskette.

To use your boot disk, your system's CMOS must be configured to allow booting from the floppy drive. If your system boots from its

hard drive even when the boot floppy is present, you must change your system's CMOS configuration. The relevant option is generally named Boot Sequence, Boot Order, or something similar. The value you want is generally labeled A:,C: or something similar. Consult your system's documentation for further information.

D.3 The lilo Loader

Most PCs can be booted from a floppy drive or hard drive; many recently manufactured PCs can be booted from a CD-ROM drive. The first sector of a disk, diskette, or partition is known as the *boot sector*. The boot sector associated with a disk or diskette (the first sector of the disk or diskette) is known as the *master boot record*. In order for a diskette or disk to be bootable, it must contain a boot loader, which can reside in:

- The boot sector of the floppy diskette.
- The master boot record (MBR) of the first hard disk or the first CD-ROM drive, if the PC supports booting from a CD-ROM diskette.
- The boot sector of a Linux file system partition on the first hard disk.
- The boot sector of an extended partition on the first hard disk.

The `lilo` loader, or simply `lilo`, is a simple boot loader that can load Linux, Microsoft Windows 3.x and 9x, and other popular operating systems. Most users install `lilo` on the MBR of their system's first hard disk. That way, when the system is started, it boots `lilo`, which can be used to load Linux, Microsoft Windows, or another operating system.

The Debian GNU/Linux installation procedure automatically installs `lilo` as part of the base system. Therefore, you don't need to install `lilo`; you merely need to configure it.

However, before configuring `lilo`, you should make a fresh startup diskette for your system. If you have more than one operating system on your hard drive, make a startup diskette for each. Then, test each startup diskette. If you have a startup diskette for each operating system on your hard drive, you'll be able to start each operating system even if the MBR becomes damaged.

You can edit `lilo`'s configuration file, `/etc/lilo.conf`, by using your favorite text editor. The `lilo` User's Guide, found in the `/usr/doc/lilo` directory, describes the format of the configuration file directives. Here's a typical `lilo.conf` file:

```
boot=/dev/hda2
root=/dev/hda2
install=/boot/boot.b
map=boot/map
vga=normal
delay=20
image=/vmlinuz
    label=Linux
    read-only
```

Table D.1 describes the directives used in the example file:

Table D.1: Commonly Used lilo Directives

Directive	Description
<code>boot=</code>	Specifies the device or partition to which the boot image is written.
<code>root=</code>	Specifies the partition which will be mounted as root when the kernel is loaded.
<code>install=</code>	Specifies the file that is used to create the boot sector. This file normally resides in <code>/boot</code> .
<code>map=</code>	Specifies the file that indicates the location of the kernel. This file normally resides in <code>/boot</code> .

Directive	Description
<code>vga=</code>	Specifies the video mode set when the kernel is loaded. Possible values are <code>normal</code> , <code>extended</code> , and <code>ask</code> , or a decimal number that gives the video mode.
<code>delay=</code>	Specifies the delay (in tenths of a second) before automatically booting the default kernel, which is the first specified in the configuration file.
<code>image=</code>	Specifies the file that contains the kernel image.
<code>label=</code>	Specifies the name by which the kernel image is known.
<code>read-only</code>	If present, specifies that the root file system will be mounted read-only.

After revising the configuration file, you run `lilo` by issuing the command:

```
/sbin/lilo
```

This causes `lilo` to update the master boot record or boot sector, according to the information in its configuration file. Then, you can boot your system according to the latest `lilo` specifications.

To boot your system, `lilo` uses your system's BIOS, which may not be able to load a Linux kernel (or other program) stored beyond cylinder 1023 of your hard drive. If you're installing Linux to a preexisting hard drive, you may not be able to place your Linux

kernel in an appropriate location. In that case, you won't be able to use `lilo` to boot your system.

D.4 The loadlin Loader

Another way of booting Linux is by using `loadlin`, an MS-DOS program that can load a Linux kernel. To load Linux, `loadlin` relies on MS-DOS rather than your system's BIOS; therefore, `loadlin` can load a kernel stored beyond cylinder 1023. More generally, it can load a kernel from any filesystem or location accessible to MS-DOS.

However, `loadlin` cannot be run from a DOS Prompt Window within Windows 3.x or 9x. You must start your system in MS-DOS mode in order for `loadlin` to work. By making the proper entries to your *config.sys* file, you can create a convenient boot menu that lets you boot MS-DOS, Windows, or Linux.

D.4.1 Installing loadlin

The `loadlin` program must have access to the file containing the Linux kernel you want to boot. The easiest way to get this file onto your Windows system is to boot Linux, make sure the Windows filesystem that corresponds to the Windows C: drive is mounted, and copy the kernel file. The following commands assume that your Windows filesystem is mounted as */mnt/c* and that you want to store the kernel in the directory *c:\linux*.

```
mkdir /mnt/c/linux
cp /boot/vmlinuz /mnt/c/linux/vmlinuz
```

The `loadlin` program needs to know the identity of your Linux root partition. To learn the name of the root partition, issue the command:

```
mount
```

The command reports all the mounted devices:

```
/dev/hda2 on / type ext2 (rw)
none on /proc type proc (rw)
/dev/hda1 on /boot type ext2 (ro)
```

```
none on /dev/pts type devpts (rw,mode=0622)
/dev/hdc on /cdrom type iso9660 (ro)
```

The root partition is the partition mounted as /. Here, it's /dev/hda2. Make a note of the partition name. Then, boot your Microsoft Windows system and copy the file *loadlin.exe* from the /dosutils directory to a convenient location on your hard drive.

D.4.2 Using loadlin

To test `loadlin`, restart your Windows system in MS-DOS mode, by clicking on Start → Shut Down, choosing Restart in MS-DOS Mode from the Shut Down Windows dialog box that appears, and clicking on OK. When the MS-DOS prompt appears, change to the directory containing `loadlin` and issue the command:

```
loadlin c:\linux\vmlinuz root=/dev/hd
xn ro
```

where /dev/hd *xn* is the root partition of your Linux system, which you earlier recorded. If your Linux files are stored in a directory other than *\linux*, you must adjust the command's first argument appropriately. Your Linux system should boot. If it does not, check your work and try again.

D.4.3 Configuring loadlin

Once you're satisfied that `loadlin` works with your system, you can configure your system to make using `loadlin` more convenient. Microsoft Windows supports a simple boot menu that will let you decide whether to boot Linux or Windows. To create such a boot menu, boot Microsoft Windows and use Notepad to add the following lines to the top of your *config.sys* file:

```
[menu]
menuitem=Linux, Boot Linux
menuitem=Win95, Boot Windows 95
menudefault=Linux, 15
```

```
[linux]
shell=c:\linux\loadlin.exe @c:\linux\bootopts.txt
```

```
[win95]
```

If you're using Windows 3.x or Windows 98 rather than Windows 95, change the file accordingly.

The *config.sys* file is located in the root directory of the C: drive. If your system has no *config.sys* file, create one using the lines given.

Now, add the following lines to the top of your *autoexec.bat* file:

```
goto %config%
:win95
```

The *autoexec.bat* file is located in the root directory of the C: drive. If your system has no *autoexec.bat* file, create one using the lines given.

Finally, use Notepad to create the file *bootopts.txt* in the \linux directory. The file should have the contents similar to the following:

```
c:\linux\vmlinux root=/dev/hd
xn ro
```

Be sure to substitute the name of your Linux root partition for the placeholder `/dev/hd xn`. You can specify additional options if you like. The next section introduces you to the most popular ones.

Now, when you boot your system, you'll see a convenient menu that lets you type a digit to choose which operating system you want to boot.

Another convenient way to use `loadlin` with Windows 95 (but not Windows 98) is to create a program shortcut that switches your system to MS-DOS mode and runs `loadlin`. Launch the dialog box for creating the shortcut by right clicking on the desktop and clicking on New → Shortcut. The dialog lets you specify the contents of the

autoexec.bat and *config.sys* files. The former should be empty and the latter should contain the line:

```
shell=c:\linux\loadlin.exe @c:\linux\bootopts.txt
```

D.5 Boot Parameters

When you boot your system by using `loadlin`, you can specify several parameters that control the boot process. Such parameters are sometimes needed to take advantage of unusual hardware devices, large amounts of RAM (more than 128 MB), and so on. Similarly, when you boot by using `lilo`, you can also boot parameters to control the boot process; you can specify `lilo`'s boot parameters by using `linuxconf` or by editing `lilo`'s configuration file. You can also specify boot parameters to `lilo` by typing them in response to `lilo`'s prompt.

Boot parameters are specified using a two-part directive that includes:

- the name of the parameter
- an optional list of options, which consists of an equals sign (=) followed by a comma-separated list of option values

No spaces may appear in the directive. As an example, the following directive specifies the identity of the Linux root partition:

```
root=/dev/hda1
```

You can specify multiple directives by separating them with a space. For example, the following specifies the identity of the Linux root partition and specifies that the root partition is initially mounted read-only, so that a thorough check of its filesystem can be performed:

```
root=/dev/hda1 ro
```

Most directives are interpreted by the kernel, though `lilo` is also capable of processing directives. If you specify a directive that neither the kernel nor `lilo` understands (assuming you're using `lilo`), a directive that includes an equals sign is passed to the `init`

process as an environment variable. You've learned about environment variables in **Chapter 13**,

Conquering the BASH Shell. A non-kernel directive that doesn't include an equals sign is passed to the `init` process. An example of this usage is specifying the directive `single`, which causes `init` to start your system in single-user mode:

```
root=/dev/hda1 ro single
```

This directive is especially useful when booting your system using `lilo`; by specifying the `single` directive in response to `lilo`'s command prompt, you can boot your system in single-user mode.

D.5.1 General Boot Arguments

[Table D.2](#) describes some of the most popular and useful boot arguments. These arguments apply to your system as a whole; in subsequent sections you'll learn about other boot arguments that apply to specific devices or functions. In addition to boot arguments previously introduced, the table describes the `reserve` argument, which is helpful in avoided system memory conflicts.

Table D.2: Selected General Boot Arguments

Argument	Description and options
<code>init=</code>	Specifies arguments passed by the kernel to the <code>init</code> process.
<code>mem=</code>	Specifies the amount of physical memory available to Linux; lets you instruct Linux to avoid high memory areas used by some systems for BIOS or caching. You can specify the amount as a hexadecimal number, or as a decimal number followed by <code>k</code> or <code>M</code> , denoting kilobytes or megabytes, respectively.

Argument	Description and options
<code>reserve=</code>	Specifies I/O ports that must not be probed. The port number is specified by using a hexadecimal number and the range is specified by using a decimal number. For example, <code>reserve=0x320, 32</code> specifies the I/O ports 320-33f must not be probed.
<code>ro</code>	Initially mount the root filesystem in read-only mode, so that a more effective filesystem check can be done.
<code>root=</code>	Identifies the root filesystem: <ul style="list-style-type: none"> • <code>/dev/fd <i>n</i></code>, floppy disk <i>n</i> (0 or 1) • <code>/dev/hd <i>xn</i></code>, partition <i>n</i> of IDE drive <i>x</i> (a to d) • <code>/dev/sd <i>xn</i></code>, partition <i>n</i> of SCSI drive <i>x</i> (a to e)
<code>rw</code>	Initially mount the root filesystem in read-write mode; do not perform a filesystem check.
<code>vga=</code>	Specifies the default display mode set before booting. Specifying <code>vga=ask</code> will cause <code>lilo</code> to list the available video modes. You can then specify the desired mode in place of ask. Note: This argument is interpreted by <code>lilo</code> and will have no effect if another loader is used.

D.5.2 RAM Disk Boot Arguments

[Table D.3](#) describes several boot arguments used in working with RAM disks. You won't likely need to specify any of these; but knowing about them may help you understand boot specifications written by others, including those used by Debian GNU/Linux.

Table D.3: Selected RAM Disk Arguments

Argument	Description and options
<code>load_ramdisk=</code>	Specifies that a ramdisk is not to be loaded (0) or is to be loaded (1).
<code>prompt_ramdisk=</code>	Specifies whether to provide a prompt instructing the user to insert a floppy containing a ramdisk (1) or provide no such prompt (0).
<code>ramdisk_size=</code>	Specifies the amount of RAM to be allocated to a ramdisk. If not specified, the default value is 4 MB.
<code>ramdisk_start=</code>	Specifies the offset (in disk blocks from the start of the boot media) of the ramdisk data; lets a kernel and ramdisk data occupy the same floppy disk.

D.5.3 SCSI Host Adapter Boot Arguments

[Table D.4](#) describes the most often used boot arguments related to SCSI host adapters. [Table D.5](#) describes the options used by the SCSI

host adapter boot arguments and other boot arguments.

Table D.4: Selected SCSI Host Adapter Arguments

Argument	Description and options
<code>advansys=</code>	Advansys SCSI host adapter: <code>iobase, [iobase, [iobase, [iobase]]]</code>
<code>aha152x=</code>	Adaptec aha151x, ada152x, aic6260, aic6360, and SB16-SCSI SCSI host adapters: <code>iobase[, irq[, scsi_id[, reconnect[, parity]]]</code>
<code>aha1542=</code>	Adaptec aha154x SCSI host adapter: <code>iobase[, buson,busoff[, dmaspeed]]</code>

Argument	Description and options
aic7xxx=	<p>Adaptec aha274x, aha284x, aic7xxx SCSI host adapters:</p> <p><i>extended,</i> <i>no_reset</i></p>
AM53C974=	<p>AMD AM53C974-based SCSI host adapters:</p> <p><i>scsi-id,</i> <i>dev_id,</i> <i>dmасpeed,</i> <i>offset</i></p> <p>See the file <i>linux/drivers/scsi/README.AM53C974</i>.</p>
buslogic=	<p>BusLogic SCSI controller:</p> <p>Many options are available. See the BootPrompt-HOWTO.</p>
eata=	<p>EATA SCSI host adapter:</p> <p><i>iobase, [</i> <i>iobase, [</i> <i>iobase, [</i> <i>iobase]]]</i></p>

Argument	Description and options
<code>fdomain=</code>	<p>Future Domain SCSI controller:</p> <p><i>iobase,</i> <i>irq[,</i> <i>scsi_id]</i></p>
<code>in2000=</code>	<p>Always SCSI host adapter:</p> <p>The driver for the Always SCSI controller accepts options in somewhat different format than other drivers. See the BootPrompt-HOWTO.</p>
<code>max-scsi-luns=</code>	<p>Specifies the maximum number of SCSI logical units to be probed; lets you avoid probing devices that might lock up the SCSI bus.</p>
<code>ncr5380=</code>	<p>NCR 5380-based SCSI host adapters:</p> <p><i>iobase,</i> <i>irq,</i> <i>dma</i></p> <p><i>membase,</i> <i>irq,</i> <i>dma</i></p>

Argument	Description and options
ncr53c400=	<p>NCR 53c400-based SCSI host adapters:</p> <pre> iobase, irq membase, irq </pre>
ncr53c406a=	<p>NCR 53c406a-based SCSI host adapters:</p> <pre> iobase, irq, pio membase, irq, pio </pre>
ppa=	<p>IOMEGA parallel port SCSI adapter for ZIP drive:</p> <pre> iobase, speed_high, speed_low, nybble </pre>
pas16=	<p>Pro Audio Spectrum SCSI host adapter:</p> <pre> iobase, irq </pre>

Argument	Description and options
<code>st0x=</code>	Seagate ST-0x SCSI host adapter: <i>membase,</i> <i>irq</i>
<code>t128=</code>	Trantor T128 SCSI host adapter: <i>membase,</i> <i>irq</i>
<code>tmc8xx=</code>	Future Domain TMC-8xx and TMC-950 SCSI host adapters: <i>membase,</i> <i>irq</i>
<code>u14-34f=</code>	Ultrastor SCSI host adapter: <i>iobase,</i> [<i>iobase,</i> [<i>iobase,</i> [<i>iobase</i>]]]
<code>wd7000=</code>	Western Digital WD7000 SCSI host adapter: <i>irq,</i> <i>dma,</i> <i>iobase</i>

For example, from [Table D.4](#) you can learn that Adaptec aha154x SCSI host adapters use a boot argument having the form:

```
iobase[,  
buson,busoff[,  
dm speed]]
```

[Table D.5](#) helps you understand the form of the *iobase* option and the other italicized options. The *iobase* option, for example, lets you specify the I/O port associated with the SCSI host adapter. Don't include the square brackets in your boot argument; they merely indicate which options must be present. For example, you can specify a boot argument for an Adaptec aha154x SCSI host adapter by writing only an *iobase* option; the remaining options are optional. However, as indicated by the square brackets, if you include a *buson* option, you must include a *busoff* option. Similarly, to include the *dm speed* option, you must include each of the other options. Here's an example of a complete boot argument:

```
aha1542=0x300,11,4
```

Table D.5: Selected Boot Prompt Options

Option	Description and options
<i>busoff</i>	The interval (number of microseconds) during which the device will relinquish the ISA bus, specified as a decimal integer; for example, 4.
<i>buson</i>	The interval (number of microseconds) during which the device will dominate the ISA bus, specified as a decimal integer; for example, 11.

Option	Description and options
<i>ctl</i>	The I/O port used for control, specified as a hexadecimal number; for example, 0x300.
<i>cyl,head,sect</i>	The geometry of the storage device, specified as three integers denoting the number of cylinders, heads, and sectors, respectively.
<i>dev_id</i>	A SCSI device with which the host adapter communicates, specified as a decimal integer; for example, 2.
<i>dma</i>	The DMA (direct memory access) channel by used by the device, specified as a decimal integer; for example, 3.
<i>dmask</i>	The rate (in MB/sec.) at which DMA transfers are performed, specified as a decimal integer; for example, 5.
<i>extended</i>	Specifies whether extended translation for large disks is enabled (1) or not (0).
<i>magic_number</i>	Specifying the value 79 causes the driver to attempt to work, even if the firmware version is unknown; other values are ignored.

Option	Description and options
<i>no_reset</i>	Specifies whether the driver should reset the SCSI bus when setting up the host adapter at boot (1) or not (0).
<i>iobase</i>	An I/O port, specified as a hexadecimal number; for example, 0x300.
<i>irq</i>	A hardware interrupt number, specified as a decimal integer; for example, 5.
<i>is_pas_card</i>	Specifies whether a Pro Audio Spectrum card is used (PAS); otherwise, do not specify this option.
<i>membase</i>	The base address of a memory region used for memory-mapped I/O, specified as a hexadecimal number; for example, 0x2000.
<i>parity</i>	Specifies whether the SCSI host adapter uses parity (1) or does not use parity (0).
<i>pio</i>	Specifies whether <i>insl</i> and <i>outsl</i> multi-byte instructions (1) or <i>inb</i> and <i>outb</i> single-byte instructions (0) are used.

Option	Description and options
<i>reconnect</i>	Specifies whether the SCSI host adapter is allowed to disconnect and reconnect (1) or holds a connection until the operation is complete (0).
<i>scsi_id</i>	The ID by which the SCSI host adapter identifies itself, specified as a decimal integer; for example, 7.

To determine a proper value for options described in [Table D.5](#), you must often know something about the hardware structure of your system. The procedures described in **Chapter 2**,

Preparing to Install Linux will help you.

D.5.4 IDE Hard Drive and CD-ROM Boot Arguments

[Table D.6](#) describes the most commonly used boot arguments associated with IDE hard drives and CD-ROM drives. Refer to [Table D.7](#) to determine the form of the italicized options.

Table D.6: Selected IDE Hard Drive Arguments

Argument	Description and Options
<code>ide0=</code>	<p>IDE hard drive or CD-ROM:</p> <ul style="list-style-type: none">• <code>ali14xx</code>, probe for, and support, the ali14xx interface.• <code>cmd640_vlb</code>, probe for, and support, the cmd640 chip (required for controllers using a VLB interface).• <code>dtc2278</code>, probe for, and support, the dtc2278 interface.• <code>ht6560b</code>, probe for, and support, the ht6560b interface.• <code>qd6580</code>, probe for, and support, the qd6580 interface.• <code>umc8672</code>, probe for, and support, the umc8672 interface.

Argument	Description and Options
hd <i>x</i> =	<p>IDE hard drive or CD-ROM (<i>x</i> denotes the physical device, and must be a letter from <i>a</i> to <i>h</i>):</p> <ul style="list-style-type: none"> • <code>autotune</code>, which specifies that the driver should attempt to tune the interface to the fastest possible mode and speed. • <code>cdrom</code>, which specifies that the drive is a CD-ROM drive. • <code>cyl, head, sect</code>, which specifies the geometry of the drive. • <code>none</code>, which specifies that the drive is not present - do not probe. • <code>noprobe</code>, which specifies that the driver should not probe for the device. • <code>nowerr</code>, which specifies that the <code>WRERR_STAT</code> bit should be ignored on this drive.

Argument	Description and Options
ide x =	<p>IDE hard drive or CD-ROM (<i>x</i> specifies the physical device, and must be a digit from 0 to 3):</p> <ul style="list-style-type: none"> • <i>autotune</i>, which specifies that the driver should attempt to tune the interface to the fastest possible mode and speed. • <i>iobase</i>, which specifies the I/O port used by the drive. • <i>iobase , ctl</i>, which specifies the I/O port and control port used by the drive. • <i>iobase , ctl , irq</i>, which specifies the I/O port, control port, and IRQ used by the drive. • <i>noautotune</i>, which specifies that the driver should not attempt to tune the interface for fastest mode and speed.
	<ul style="list-style-type: none"> • <i>noprobe</i>, which specifies that the driver should not probe for the device. • <i>serialize</i>, which specifies that I/O operations should not be overlapped.

D.5.5 Non-IDE CD-ROM Drive Boot Arguments

[Table D.7](#) describes the most common boot arguments for non-IDE CD-ROM drives. Refer to [Table D.5](#) to determine the form of the italicized options.

Table D.7: Selected CD-ROM Arguments

Argument	Description and Options
aztcd=	<p>Aztech CD-ROM:</p> <pre><i>iobase</i>[, <i>magic_number</i>]</pre>
cdu31a=	<p>Sony CDU-31A or CDU-33A CD-ROM:</p> <pre><i>iobase</i>, [<i>irq</i>[, <i>is_pas_card</i>]]</pre>
sonycd535=	<p>Sony CDU-535 CD-ROM:</p> <pre><i>iobase</i>[, <i>irq</i>]</pre>
gsacd=	<p>Goldstar CD-ROM:</p> <pre><i>iobase</i></pre>
isp16=	<p>ISP16 CD-ROM:</p> <pre>[<i>port</i>[, <i>irq</i>[, <i>dma</i>]]][[,] <i>drive_type</i>]</pre>

Argument	Description and Options
mcd=	<p>Mitsumi CD-ROM:</p> <pre><i>iobase</i>, [<i>irq</i> [, <i>wait_value</i>]]</pre>
optcd=	<p>Optical Storage CD-ROM:</p> <pre><i>iobase</i></pre>
cm206=	<p>Phillips CD206 CD-ROM:</p> <pre>[<i>iobase</i>] [, <i>irq</i>]</pre>
sjcd=	<p>Sanyo CD-ROM:</p> <pre><i>iobase</i> [, <i>irq</i> [, <i>dma_channel</i>]]</pre>
sbpcd=	<p>SoundBlaster Pro CD-ROM:</p> <pre><i>iobase</i>, <i>type</i></pre>

D.5.6 Floppy Drive Boot Arguments

A few systems require special boot arguments to make best use of their floppy drives. [Table D.8](#) describes the most common boot arguments related to floppy drives. The file *README.fd* in *linux/drivers/block* describes additional arguments. Floppy drives that are not well behaved may malfunction if you specify the `daring` option, which you should use only with care.

Table D.8: Selected Floppy Disk Arguments and Options

Argument and Option	Description
<code>floppy=asus_pci</code>	Specifies that only units 0 and 1 are allowed, to work around problem with BIOS of certain ASUS motherboards.
<code>floppy=daring</code>	Specifies that the floppy controller is well behaved, allowing more efficient operation.
<code>floppy=0,daring</code>	Specifies that the floppy controller may not be well behaved (default).
<code>floppy=thinkpad</code>	Specifies that the system is an IBM Thinkpad.

Argument and Option	Description
<code>floppy=no_unexpected_interrupts</code> or <code>floppy=L40SX</code>	Specifies that a message should be printed when an unexpected interrupt is received. This is required by IBM L40SX laptops in certain video modes.

D.5.7 Bus Mouse Boot Arguments

Two boot arguments provide bus mouse support. The first supports the Microsoft bus mouse:

```
msmouse=  
irq
```

The second supports any non-Microsoft bus mouse:

```
bmouse=  
irq
```

Each argument accepts a single option specifying the IRQ associated with the mouse.

D.5.8 Parallel Port Printer Boot Arguments

The Linux printer driver claims all available parallel ports. If you want to access a device other than a printer attached to a parallel port, you must instruct the printer driver to reserve only the ports associated with printers. To do so, use the `lp` boot argument, which takes as its options a list of ports and IRQs use to support printers. For example, the following boot argument specifies two printers:

```
lp=0x3bc,0,0x378,7
```

The first printer is on port `0x3bc` and the second is on port `0x378`. The first printer uses a special IRQ-less mode known as polling, so its IRQ is specified as 0. The second printer uses IRQ 7.

To disable all printers, specify `lp=0`.

D.6 Using Loadable Ethernet Drivers

Early versions of Linux used a so-called monolithic kernel. At that time, Linux distributions typically included several kernels, offering support for a variety of devices that might be needed to boot and install a Linux system. Devices not needed to boot and install a system - so-called special devices - had second-class status. To access special devices, users had to compile customized kernels that included support for those devices. When a user added a device to a system, it was often necessary to compile a new kernel, which was something of an inconvenience.

More recent versions of Linux feature a modular kernel, which allows drivers to be dynamically loaded on command. This makes it much easier than before to configure your Linux system to support Ethernet cards and other special devices. Debian GNU/Linux is generally able to configure your primary Ethernet card automatically, by probing for it during installation of Linux.

However, the autoprobe doesn't always succeed. Moreover, if you have more than one Ethernet card, the installation program sets up only the first card it finds. To set up additional cards, you need to know a bit about Linux's loadable modules.

D.6.1 Dynamically Loading a Modular Driver

To dynamically load a modular driver, you issue the following command:

```
insmod  
driver
```

Where *driver* specifies the module to be loaded. [Table D.9](#) lists modular Ethernet drivers and the cards they support. As an example, the command:

```
insmod ne2k-pci
```


loads the modular driver for the PCI-based NE2000 Ethernet card.

Table D.9: Modular Ethernet Drivers and Supported Cards

Driver	Cards
3c501	3Com 3c501
3c503	3Com EtherLink II
3c505	3Com EtherLink Plus
3c507	3Com EtnerLink 16
3c509	3Com EtherLink III
3c515	3Com EtherLink XL
8390	National Semiconductor NS8390
a2065	Commodore/Ameristar A2065
ac3200	Ansel Communications AC3200 (EISA)
apricot	Apricot 82596

Driver	Cards
arcnet	ARCnet COM9026 and COM20020
ariadne	Village Tronic Ariadne
at1700	Allied Telesis AT1700
atari_bionet	Atari BIONET-100
atari_pamsnet	Atari PAMsNet
atarilance	Atari VME Lance
cops	LocalTalk PC
de4x5	EtherWORKS DE425, DE434, DE435, DE450, DE500, DC21040, DC21041, DC21142, DC21143
de600	D-Link DE-600
de620	D-Link DE-620

Driver	Cards
depca	DEPCA/EtherWORKS DEPCA, DE100, DE101, DE200, DE201, DE202, DE210, DE422
dgrs	Digi RightSwitch SE-X
e2100	Cabletron E2100
eeepro	Intel EtherExpress Pro/10
eeepro100	Intel EtherExpress Pro/100
eeexpress	Intel EtherExpress
epic100	SMC 83c170 EPIC/100
eth16i	ICL EtherTeam 16i, EtherTeam 32 (EISA)
ewrk3	EtherWORKS DE203, DE204, DE205
hp-plus	HP PCLAN/Plus
hp	HP LAN

Driver	Cards
hp100	HP J2585A, J2585B, J2970, J2973, J2573 Compex ReadyLink ENET100-VG4 Compex FreedomLine 100/VG
hydra	Hydra Amiganet
ibmtr	IBM token ring
lance	Allied Telesis AT1500 HP J2405 ANE 2100, 2500
ne	NE1000, NE2000 (non-PCI)
ne2k-pci	NE2000 (PCI)
ni52	Rascal-Interlan NI5210
ni65	Rascal-Interlan NI6510
pcnet	AMD PCnet32- and PCnetPCI-based cards

Driver	Cards
rt18139	<p>Cards based on the RTL8129 and RTL8139 PCI Ethernet chips, such as:</p> <p>ALFA GFC2206</p> <p>Allied Telesyn AT2550</p> <p>Genius GF100TXR (RTL8139)</p> <p>NDC Communications NE100TX-E</p> <p>SiS 900 (PCI)</p> <p>SMC 1211TX (PCI)</p>
smc-ultra	SMC Ultra, UltraEZ, Ultra32
smc9194	SMC 9000
tlan	Various Compaq and Olicom cards

Driver	Cards
tulip	<p>Cards based on the DEC 21040/21041/21140/21142/21143, such as:</p> <p>Accton EtherDuo PCI, EN1207</p> <p>Adaptec ANA6901/C, ANA6911/TX</p> <p>C-NET CNE-935</p> <p>Cogent EM100, EM110, EM400, EM960, EM964 Quartet</p> <p>Danpex EN-9400P3</p> <p>D-Link DFE500-Tx, DE-530CT, DFE-540TX</p> <p>Linksys EtherPCI</p> <p>Kingston EtherX KNT40T, EtherX KNE100TX</p> <p>Netgear FX310 TX 10/100</p> <p>SMC EtherPower, 8432BT, EtherPower10/100, EtherPower10/100</p> <p>Surecom EP-320X</p>
tulip (cont.)	<p>Thomas Conrad TC5048</p> <p>Znyx ZX312 EtherAction, ZX314, ZX315 EtherArray, ZX342, ZX344, ZX345, ZX346, ZX348, ZX351</p>

Driver	Cards
wavelan	AT&T/NCR GIS WaveLAN
wd	WD8003, WD8013
yellowfin	Packet Engines G-NIC

When a driver is loaded, it generally probes to locate the supported device. In case an autoprobe fails, most drivers let you specify the I/O port and IRQ by using a command like the following:

```
insmod ne2k=pci io=0x280 irq=11
```

Some cards support additional options; these are documented in the file */usr/src/linux/Documentation/networking/net-modules.txt*.

D.6.2 Loading Modular Drivers at Boot Time

The Linux kernel automatically loads modules specified in the module configuration file, */etc/conf.modules*. So, once you've determined the proper module and options required by your Ethernet card, you can add a line or two to the module configuration file so that your card will be made ready to operate each time you boot your system.

The `alias` directive associates a logical module name with an actual module. Logical module names specify types of devices; for example, `eth0` specifies the first Ethernet card in a system and `eth1` specifies the second Ethernet card in a system. Suppose your system includes two Ethernet cards: a non-PCI-based NE2000 and an SMC EtherPower, which is based on DEC's TULIP chip. You could use the

following directives to automatically load these modules at boot time:

```
alias eth0 ne
alias eth1 tulip
```

If a driver requires options, you can specify them by using an `options` directive, which has the following form:

```
options
driver argument=
value[,
value,...]

argument=
value[,
value,...] ...
```

For example, you might specify the I/O port and IRQ used by the NE2000 card like this:

```
options ne io=0x280 irq=12
```

Most ISA modules accept parameters like `io=0x340` and `irq=12` on the `insmod` command line. You should supply these parameters to avoid probing for the card. Unlike PCI and EISA devices, ISA devices sometimes cannot be safely auto-probed.

D.6.3 Administering Modular Drivers

The `lsmod` command, which takes no arguments, lists the loaded modular drivers. To unload a modular driver, specify the driver as the argument of the `rmmod` command. For example, to remove the `ne` driver, issue the command:

```
rmmod ne
```

By specifying the `-a` argument, you can cause `rmmod` to unload every unused module; that is, every module not associated with an operational device:


```
rmmod -a
```

You can't remove a module that's in use; therefore, you must shut down the device before removing it. To shut down an Ethernet device, you can use `linuxconfig`. Or, you can issue the following command:

```
ifconfig eth  
n down
```

where `eth n` specifies the logical device (for example, `eth0` or `eth1`).



Learning Debian GNU/Linux

By Bill McCarty

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

E. Linux Command Quick Reference

The following list describes some of the most useful and popular Linux commands. Consult the man page for each command to learn about additional arguments and details of operation.

`adduser userid`

Creates a new `userid`, prompting for necessary information (requires `root` privileges).

`apropos keyword`

Searches the manual pages for occurrences of the specified `keyword` and prints short descriptions from the beginning of matching manual pages.

`at time`
`at -f file time`

Executes commands entered via `stdin` (or, by using the alternative form, the specified file) at the specified time. The time can be specified in a variety of ways; for example, in hour and minute format `hh: mm` or in hour, minute, month, day, and year format `hh: mm mm/ dd/ yy`.

`atq`

Prints descriptions of jobs pending via the `at` command.

`atrm job`

Cancels execution of a job scheduled via the `at` command. Use the `atq` command to discover the identities of scheduled jobs.

```
bg
bg jobs
```

Places the current job (or, by using the alternative form, the specified jobs) in the background, suspending its execution so that a new user prompt appears immediately. Use the `jobs` command to discover the identities of background jobs.

```
cal month year
```

Prints a calendar for the specified month of the specified year.

```
cat files
```

Prints the contents of the specified files.

```
cd
cd directory
```

Changes the current working directory to the user's home directory or the specified directory.

```
chgrp group files
chgrp -R group files
```

Changes the group of the specified files to the specified group. The alternative form of the command operates recursively, changing the group of subdirectories and files beneath a specified directory. The group must be named in the `/etc/groups` file, maintained by the `newgroup` command.

```
chmod mode files
chmod -R mode files
```

Changes the access mode of the specified files to the specified mode. The alternative form of the command operates recursively, changing the mode of subdirectories and files beneath a specified directory.

```
chown userid files
chown -R userid files
```

Changes the owner of the specified files to the specified userid. The alternative form of the command operates recursively, changing the owner of subdirectories and files beneath a specified directory

```
clear
```

Clears the terminal screen.

```
cmp file1 file2
```

Compares two files, reporting all discrepancies. Similar to the `diff` command, though the output format differs.

```
cp file1 file2  
cp files directory  
cp -R files directory
```

Copies a file to another file or directory, or copies a subdirectory and all its files to another directory.

```
date  
date date
```

Displays the current date and time or changes the system date and time to the specified value, of the form *MMddhhmmyy* or *MMddhhmmyyyy*.

```
df
```

Prints the amount of free disk space on each mounted filesystem.

```
diff file1 file2
```

Compares two files, reporting all discrepancies. Similar to the `cmp` command, though the output format differs.

```
dmesg
```

Prints the messages resulting from the most recent system boot.

```
du
du directories
```

Prints the amount of disk space used by the current directory (or the specified directories) and its (their) subdirectories.

```
echo string
echo -n string
```

Prints the specified text on the standard output stream. The *-n* option causes omission of the trailing newline character.

```
fdformat device
```

Formats the media inserted in the specified floppy disk drive. The command performs a low-level format only; it does not create a filesystem. To create a filesystem, issue the `mkfs` command after formatting the media.

```
fdisk device
```

Edits the partition table of the specified hard disk.

```
fg
fg jobs
```

Brings the current job (or the specified jobs) to the foreground.

```
file files
```

Determines and prints a description of the type of each specified file.

```
find path -name pattern -print
```

Searches the specified path for files with names matching the specified pattern (usually enclosed in single quotes) and prints their names. The `find` command has many other arguments and functions; see the online documentation.

```
finger users
```

Prints descriptions of the specified users.

```
free
```

Displays the amount of used and free system memory.

```
ftp hostname
```

Opens an FTP connection to the specified host, allowing files to be transferred. The FTP program provides subcommands for accomplishing file transfers; see the online documentation.

```
grep pattern files  
grep -i pattern files  
grep -n pattern files  
grep -v pattern files
```

Search the specified files for text matching the specified pattern (usually enclosed in single quotes) and print matching lines. The *-i* option specifies that matching is performed without regard to case. The *-n* option specifies that each line of output is preceded by the file name and line number. The *-v* option reverses the matching, causing non-matched lines to be printed.

```
gzip files  
gunzip files
```

Compress (or expand) the specified files. Generally, a compressed file has the same name as the original file, followed by *.gz*.

```
head files
```

Prints the first several lines of each specified file.

```
hostname  
hostname name
```

Displays (or sets) the name of the host.

```
info
```

Launches the GNU Texinfo help system.

```
init run_level
```

Changes the system run level to the specified value (requires `root` privileges).

```
insmod module
```

Dynamically loads the specified module (requires `root` privileges).

```
jobs
```

Displays all background jobs.

```
ispell files
```

Checks the spelling of the contents of the specified files.

```
kill process_ids  
kill - signal process_ids  
kill -l
```

Kills the specified processes, sends the specified processes the specified signal (given as a number or name), or prints a list of available signals.

```
killall program  
killall - signal program
```

Kills all processes that are instances of the specified program or sends the specified signal to all processes that are instances of the specified program.

```
ln old new  
ln -s old new
```

Creates a hard (or soft) link associating a new name with an existing file or directory.

`locate pattern`

Locates files with names containing the specified pattern. Uses the database maintained by the `updatedb` command.

`lpq`

Prints the entries of the print queue.

`lpr files`

Prints the specified files.

`lprm job`

Cancels printing of the specified print queue entries. Use `lpq` to determine the contents of the print queue.

`ls`

`ls files`

`ls -a files`

`ls -l files`

`ls -lR files`

Lists (non-hidden) files in the current directory or the specified files or directories. The `-a` option lists hidden files as well as non-hidden files. The `-l` option causes the list to include descriptive information, such as file size and modification date. The `-R` option recursively lists the subdirectories of the specified directories.

`mail`

Launches a simple mail client that permits sending and receiving email messages.

`man title`

`man section title`

Prints the specified man page.

```
mkdir directories  
mkdir -p directories
```

Creates the specified directories. The *-p* option causes creation of any parent directories needed to create a specified directory.

```
mkfs -t type device
```

Creates a file system of the specified type (such as `ext2` or `msdos`) on the specified device (requires `root` privileges).

```
mkswap device
```

Creates a Linux swap space on the specified hard disk partition (requires `root` privileges).

```
more file
```

Lets the user peruse a file too large to be displayed as a single screen (page) of output. The `more` command provides many subcommands that let the user navigate the file. For example, the **S**pace key moves forward one page, the **b** key moves back one page, and the **q** key exits the program.

```
mount  
mount device directory  
mount -o option -t type device directory
```

Prints the mounted devices or mounts the specified device at the specified mount point (generally a subdirectory of `/mnt`). The `mount` command consults `/etc/fstab` to determine standard options associated with a device. The command generally requires `root` privileges. The *-o* option allows specification of a variety of options; for example, `ro` for read-only access. The *-t* option allows specification of the filesystem type (for example, `ext2`, `msdos`, or `iso9660`, the filesystem type generally used for CD-ROMs).

```
mv paths target
```

Moves the specified files or directories to the specified target.

```
newgroup group
```

Creates the specified group.

```
passwd  
passwd user
```

Changes the current user's password, or that of the specified user (requires `root` privileges). The command prompts for the new password.

```
ping host
```

Sends an echo request via TCP/IP to the specified host. A response confirms that the host is operational.

```
pr files
```

Formats the specified files for printing, by inserting page breaks and so on. The command provides many arguments and functions.

```
ps  
ps -Aux
```

Displays the processes associated with the current userid or displays a description of each process.

```
pwd
```

Prints the absolute path corresponding to the current working directory.

```
reboot
```

Reboots the system (requires `root` privileges).

```
reset
```

Clears the terminal screen and resets the terminal status.

```
rm files
rm -i files
rm -f files
rm -if files
rm -rf files
```

Deletes the specified files or (when the *-r* option is specified) recursively deletes all subdirectories of the specified files and directories. The *-i* option causes the command to prompt for confirmation; the *-f* option suppresses confirmation. Because deleted files cannot generally be recovered, the *-f* option should be used only with extreme care, particularly when used by the `root` user.

```
rmdir directories
rmdir -p directories
```

Deletes the specified empty directories or (when the *-p* option is specified) the empty directories along the specified path.

```
shutdown minutes
shutdown -r minutes
```

Shuts down the system after the specified number of minutes elapses (requires `root` privileges). The *-r* option causes the system to be rebooted once it has shut down.

```
sleep time
```

Causes the command interpreter to pause for the specified number of seconds.

```
sort files
```

Sorts the specified files. The command has many useful arguments; see the online documentation.

```
split file
```

Splits a file into several smaller files. The command has many arguments; see the online documentation.

```
su
su user
su -
su - user
```

Changes the current userid to `root` or to the specified userid (the latter requires `root` privileges). The `-` option establishes a default environment for the new userid.

```
swapon device
```

Enables use of the specified device for swapping (requires `root` privileges).

```
swapoff device
```

Disables use of the specified device for swapping (requires `root` privileges).

```
sync
```

Completes all pending input/output operations (requires `root` privileges).

```
tail file
tail -n file
tail -f file
```

Prints the last several lines of the specified files. The `-n` option specifies the number of lines to be printed. The `-f` option causes the command to continuously print additional lines as they are written to the file.

```
talk user
```

Launches a program that allows a chat-like dialog with the specified user.

```
tar cvf tar_file files
tar zcvf tar_file files
```

Creates a tar file with the specified name, containing the specified files and their subdirectories. The `z` option specified that the tar file will be compressed.

```
tar xvf tar_file
tar zxvf tar_file
```

Extracts the contents of the specified tar file. The `z` option specified that the tar file has been compressed.

```
telnet host
```

Opens a login session on the specified host.

```
top
```

Prints a display of system processes that's continually updated until the user presses the `q` key.

```
traceroute host
```

Uses echo requests to determine and print a network path to the host.

```
umount device
```

Unmounts the specified filesystem (generally requires `root` privileges).

```
uptime
```

Prints the system uptime.

```
w
```

Prints the current system users.

```
wall
```

Prints a message to each user except those who've disabled message reception. Type **Ctrl-D** to end the message.

`wc files`

Prints the number of characters, words, and lines in the specified files.

[Table E.1](#) identifies Linux commands that perform functions similar to MS-DOS commands. The operation of the Linux command is not generally identical to that of the corresponding MS-DOS command. See the index to this book or the Linux online documentation for further information about Linux commands.

Table E.1: MS-DOS Commands and Related Linux Commands

MS-DOS	Linux
ATTRIB	chmod
CD	cd
CHKDSK	df, du
DELTREE	rm -R
DIR	ls -l
DOSKEY	(built-in; no need to launch separately)

MS-DOS	Linux
EDIT	ae, vi, and so on
EXTRACT	tar
FC	cmp, diff
FDISK	fdisk
FIND	grep
FORMAT	fdformat
MORE	more
MOVE	mv
SORT	sort
START	at, bg
XCOPY, XCOPY32	cp



Learning Debian GNU/Linux

By Bill McCarty

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

F. Open Publication License

Contents:

I. REQUIREMENTS ON BOTH UNMODIFIED AND MODIFIED
VERSIONS

II. COPYRIGHT

III. SCOPE OF LICENSE

IV. REQUIREMENTS ON MODIFIED WORKS

V. GOOD-PRACTICE RECOMMENDATIONS

Open Publication License, Draft v0.4, 8 June 1999

F.1 I. REQUIREMENTS ON BOTH UNMODIFIED AND MODIFIED VERSIONS

The Open Publication works may be reproduced and distributed in whole or in part, in any medium physical or electronic, provided that the terms of this license are adhered to, and that this license or an incorporation of it by reference (with any options elected by the author(s) and/or publisher) is displayed in the reproduction.

Proper form for an incorporation by reference is as follows:

Copyright (c) <year> by <author's name or designee>. This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, vX.Y or later (the latest version is presently available at <http://www.opencontent.org/openpub/>).

The reference must be immediately followed with any options elected by the author(s) and/or publisher of the document (see section VI).

Commercial redistribution of Open Publication-licensed material is permitted. Any publication in standard (paper) book form shall require the citation of the original publisher and author. The publisher and author's names shall appear on all outer surfaces of the book. On all outer surfaces of the book the original publisher's name shall be as large as the title of the work and cited as possessive with respect to the title.

F.2 II. COPYRIGHT

The copyright to each Open Publication is owned by its author(s) or designee.

F.3 III. SCOPE OF LICENSE

The following license terms apply to all Open Publication works, unless otherwise explicitly stated in the document.

Mere aggregation of Open Publication works or a portion of an Open Publication work with other works or programs on the same media shall not cause this license to apply to those other works. The aggregate work shall contain a notice specifying the inclusion of the Open Publication material and appropriate copyright notice.

SEVERABILITY.

If any part of this license is found to be unenforceable in any jurisdiction, the remaining portions of the license remain in force.

NO WARRANTY.

Open Publication works are licensed and provided "as is" without warranty of any kind, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose or a warranty of non-infringement.

F.4 IV. REQUIREMENTS ON MODIFIED WORKS

All modified versions of documents covered by this license, including translations, anthologies, compilations and partial documents, must meet the following requirements.

1. The modified version must be labeled as such.
2. The person making the modifications must be identified and the modifications dated.
3. Acknowledgement of the original author and publisher if applicable must be retained according to normal academic citation practices.
4. The location of the original unmodified document must be identified.
5. The original author's (or authors') name(s) may not be used to assert or imply endorsement of the resulting document without the original author's (or authors') permission.

F.5 V. GOOD-PRACTICE RECOMMENDATIONS

In addition to the requirements of this license, it is requested from and strongly recommended of redistributors that:

1. If you are distributing Open Publication works on hardcopy or CD-ROM, you provide email notification to the authors of your intent to redistribute at least thirty days before your manuscript or media freeze, to give the authors time to provide updated documents. This notification should describe modifications, if any, made to the document.
2. All substantive modifications (including deletions) be either clearly marked up in the document or else described in an attachment to the document.

Finally, while it is not mandatory under this license, it is considered good form to offer a free copy of any hardcopy and CD-ROM expression of an Open Publication-licensed work to its author(s).



Learning Debian GNU/Linux

By Bill McCarty

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

Glossary

Absolute Path

See Path.

Access Mode

An attribute of a file or directory, which determines what operations a user may perform on the file or directory.

Alias

An alternative name for a command.

Argument

A parameter that controls the operation of a program or command.

Background

A background program is temporarily suspended from execution and does not interact with the user. See Foreground.

BIOS (Basic Input/Output System)

The program built into a computer to control its operation, especially the booting of an operating system. Most computers let the user configure various BIOS options by means of a special screen or set of screens.

Boot diskette

A diskette that contains the parts of an operating system needed to start the operating system.

Boot Sector

A sector that contains a loader program for starting an operating system.

Browser

A client program that operates under user control, especially a web client.

Client

A program that makes a request (generally via a network) of a server.

Command Interpreter

A program that accepts commands and executes (interprets) them.

Daemon

A program that runs in the background; that is, without user interaction.

Desktop

A work environment provided by a graphical user interface, generally including a video monitor background, a screen saver, and one or more taskbars and icons.

Distribution

A combination of a Linux kernel, a suite of UNIX-like command programs, and other software for installing and maintaining a Linux system.

DNS (Domain Name Server)

A computer that translates hostnames to IP addresses on behalf of requesting clients.

Dotted Quad Notation

A form of representing a 32-bit IP address, consisting of 4 numbers from 0 to 255, each separated from the others by a dot.

EIDE (Enhanced Integrated Drive Electronics)

An incremental improvement of the IDE standard for hard drives, designed to better accommodate large capacity drives.

Ethernet

A standard for sending data packets across networks, focused on the electronic signaling issues.

Foreground

A foreground program runs and interacts with the user. See Background.

FTP (File Transfer Protocol)

A protocol for transferring data files across a TCP/IP network.

GNU

GNU stands for "GNU's not Unix," and refers generally to software distributed under the GNU Public License (GPL).

GPL (GNU Public License)

The GNU Public License provides for free access to software published under its terms. Users are allowed to copy, modify, and redistribute GPL software.

Graphical User Interface (GUI)

A graphical user interface is a program that lets the user interact with a computer system in a highly visual manner, with a minimum of

typing. Graphical user interfaces usually require a high-resolution display and a pointing device, such as a computer mouse.

Hidden File

A file having a name that begins with a dot (.). Such files are not listed by the `ls` command unless a special argument (*-a*) is specified.

Home Directory

A directory provided for the personal files and directories of a user.

Host

A computer attached to a network.

Hostname

A name by which a host is known to other hosts on a network.

HTML

Hypertext Markup Language is the form in which web documents are transmitted and interpreted by browsers.

IDE

A popular standard for internal hard drives and CD-ROM drives of IBM-compatible systems.

Internet

A relatively loose federation of computer networks that permits data to be widely transferred among computers.

IP Number

A number that identifies a host, corresponding to a network interface associated with the host.

Kernel

The part of an operating system that contains the most primitive functions upon which other, more sophisticated functions depend.

Kill

To terminate a process.

LILO

A program often used to load the Linux kernel from a hard drive or boot diskette.

Man Page

A document that describes a Unix command or file, readable by using the `man` command.

Master Boot Record (MBR)

The first sector of a hard drive, which by convention contains a loader program for starting an operating system.

Mount

To make a filesystem available for use.

Operating System

A program that provides a user interface and an application interface (which makes it possible for application programs to run) and manages computer system resources.

Option

A command argument that takes one of a small number of values. Command arguments that specify files (for example) are not options.

Package

A file that contains a set of related files that can be installed as a unit.

Partition

An area of a hard disk, generally allocated to a specific operating system (though perhaps usable by multiple operating systems).

Path

A path denotes the location of a file or directory. The path is an absolute path if it gives the complete path, beginning with the root directory and including every subdirectory. Otherwise, the path is a relative path.

PPP

Point-to-point protocol, the most popular way of connecting a computer to the Internet via a dialup modem.

Process

An instance of a running program.

Prompt

A character or series of characters displayed by a command interpreter to inform the user that execution of a command has been completed and the interpreter is ready to accept a new command.

Relative Path

See Path.

root

The specially privileged userid used to perform Unix system administration.

Root Directory

The unique directory that has no parent directory. All other directories are children of the root directory or its subdirectories.

Route

A path along which data packets move from host to host across a network.

Run Level

The operating mode of a UNIX system; for example, single-user, multi-user without networking, or multi-user with networking.

Script

A series of commands, stored in a file for subsequent or repeated execution.

SCSI

A popular standard for internal and external hard drives and other peripherals.

Search Path

A series of directories automatically searched by a command interpreter in order to locate the program file that corresponds to a command to be executed.

Server

A program that responds to client requests, which are generally transmitted over a network.

Shell

A command interpreter.

Swap File

A disk file or partition used to temporarily store information when system memory runs low.

Symbolic Link

A filesystem entity that lets you associate an alternative name with a file or directory.

System Administrator

The user who installs, configures, and otherwise maintains the software (and possibly the hardware) associated with a computer system.

TCP/IP (Transmission Control Protocol/Internet Protocol)

A standard method of sending data packets across a computer network, focused on the routing and connection issues.

Terminal

A combination of a keyboard and monitor, which together provide the capability to interact with a computer system.

Text Editor

A program that lets you create and modify the contents of text files.

Telnet

A protocol for establishing a login session via TCP/IP on a remote system.

Userid

The unique identifier associated with a system user.

Window Manager

A program that manages a graphical user interface, determining the appearance of windows (by providing standard elements such as title bars, for example) and determining the response to operations such as clicking on the desktop.

Working Directory

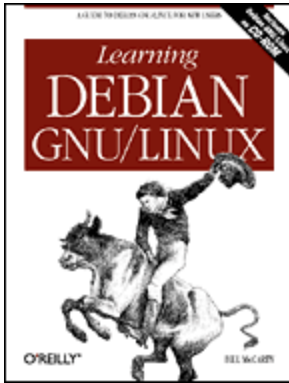
The directory that is implicitly combined with a relative path reference to determine the corresponding absolute path reference.

X Server

A program that implements X for some platform and type of video hardware.

X

A sophisticated and powerful graphical user interface implemented on a variety of computer platforms.



Learning Debian GNU/Linux

By [Bill McCarty](#)

1st Edition September 1999

1-56592-705-2, Order Number: 7052

360 pages, \$34.95 , Includes CD-ROM

[Symbols](#) | [A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Z](#)

Index: Symbols and Numbers

- (#) character : [13.2.1. Typing Shell Commands](#)
- (#) comment token : [12.2.1.3. The httpd.conf File](#)
- (#) hash mark : [7.2.1. Configuring Local Drives](#)
- (#) pound sign : [3.1.4.5. Exiting dselect](#)
- (\$) dollar sign
 - [3.1.4.5. Exiting dselect](#)
 - [13.2.7. Shell Variables](#)
- (&) ampersand : [13.3.4. Periscope: A Useful Networking Script](#)
- (&) character : [13.2.1. Typing Shell Commands](#)
- (") double quote : [13.2.9. Quoted Strings](#)
- (') single quote
 - [4.3.4.4. Creating a directory.](#)
 - [13.2.9. Quoted Strings](#)
- (*) asterisk : [4.3.5.5. Finding a file](#)
- (,) comma, user accounts and : [7.1.4.4. Removing a member from a group](#)
- (.) dot : [4.3.4.3. Displaying directory contents](#)
- (.) dot character : [7.1.1. Creating a User Account](#)
- (.) dot in filename : [4.3.4.4. Creating a directory.](#)
- (.) single dot : [4.3.3.3. Absolute and relative pathnames](#)
- (..) two dots : [4.3.3.3. Absolute and relative pathnames](#)
- (/) forward slash
 - [4.3.3.2. The directory tree](#)
 - [4.3.4.4. Creating a directory.](#)
- (/) slash
 - [C.4.3. Choosing Packages for Installation or Removal](#)
 - [C.5.1. Configuring the sources.list File](#)
- (//) slashes, pair of : [C.5.1. Configuring the sources.list File](#)
- (:) colon
 - [7.1.4.4. Removing a member from a group](#)
 - [13.2.8. The Search Path](#)
- (;) character : [13.2.1. Typing Shell Commands](#)
- (\) backslash : [13.2.9. Quoted Strings](#)
- (\) backward slash : [4.3.3.2. The directory tree](#)

(_) underscore : [4.3.4.4. Creating a directory](#)
(`) back quote : [13.2.9. Quoted Strings](#)
(GMT) Universal Time : [3.1.2.16. Configuring the base system](#)
({}) curly braces : [13.2.7. Shell Variables](#)
(|) pipe redirector
 [4.3.4.3. Displaying directory contents](#)
 [13.2.6. Input/Output Redirection and Piping](#)
/etc/apt directory : [C.5.1. Configuring the sources.list File](#)
/home/httpd/html directory : [12.2.1.1. The access.conf file](#)
/root/periscope file : [13.3.4. Periscope: A Useful Networking Script](#)
/sbin directory : [4.1.7. Shutting Down the System](#)
80386/80486 processors : [2.1.1. Central Processing Unit \(CPU\)](#)
(.) dot : [3.1.2.14. Configuring the network](#)

Index: A

absolute parameters : [4.3.5.7. Working with compressed files](#)

absolute path

[4.3.3.3. Absolute and relative pathnames](#)

[4.3.4.4. Creating a directory](#)

[4.3.5.10. Running programs](#)

[13.2.5. Shell Scripts](#)

accelerated servers : [5.3. Configuring X](#)

access methods for packages : [C.4.1. Choosing the Access Method](#)

access modes

[4.3.4.3. Displaying directory contents](#)

[4.3.5.9. Working with file permissions](#)

[13.2.5. Shell Scripts](#)

access permissions : [4.3.5.9. Working with file permissions](#)

Access step (dselect program) : [3.1.4. Installing the Applications](#)

access.conf file : [12.2.1.1. The access.conf file](#)

accessing the system : [4.1.2. Logging In](#)

accounts

root user account : [3.1.3.2. Establishing a normal user account](#)

user accounts : [3.1.3.2. Establishing a normal user account](#)

deleting : [7.1.5. Deleting a User Account](#)

activating swap partitions : [3.1.2.10. Initializing and activating a swap partition](#)

active connection : [10.3.3. Viewing Samba Server Status](#)

active file shares, viewing : [10.3.3. Viewing Samba Server Status](#)

active printer shares, viewing : [10.3.3. Viewing Samba Server Status](#)

administering Linux : [7. Configuring and Administering Linux](#)

ae text editor : [C.5.1. Configuring the sources.list File](#)

aliases : [12.2.1.2. The srm.conf File](#)

shells : [13.2.4. Shell Aliases](#)

removing/preserving : [13.2.4. Shell Aliases](#)

Alt (Meta) key : [C.6.6.4. Installing the minicom package](#)

AMD processors : [2.1.1. Central Processing Unit \(CPU\)](#)

ampersand (&) : [13.3.4. Periscope: A Useful Networking Script](#)

Angoss SmartWare : [8.1.4. Other Desktop Applications](#)

anonymous FTP : [12.1. An FTP Server](#)

Apache

(see also [web server](#))

[12.1.1. Testing the FTP Server](#)

binding web servers : [12.2.1.3. The httpd.conf File](#)

installing : [C.6.7.1. Installing the apache package](#)

log files for : [7.4. Viewing System Messages and Logs](#)

resources for further information : [12.2.1. Configuration](#)

APIs (application programming interfaces) : [8.2.2. WINE](#)

applications

(see also [packages](#))

[3.1.3.6. Choosing Packages](#)

desktop : (see entries at desktop)

GNOME-compliant, appearance of : [6.7.3. The GNOME Control Center](#)

launched automatically : [6.7.3. The GNOME Control Center](#)

Applix : [8.1.1. Applix Applixware](#)

Applixware : [8.1.1. Applix Applixware](#)

installing : [C.6.3.2. Installing Applix](#)

running : [8.1.1.1. Running Applixware](#)

apt-get package management tool : [C.5. Using apt-get](#)

archives : [4.3.5.7. Working with compressed files](#)

arguments

[4.2.1. Command Structure](#)

[13.2.2. Commands and Arguments](#)

[13.3.3.7. The break and continue commands](#)

aliases and : [13.2.4. Shell Aliases](#)

commands and : [13.3.3.1. The test command](#)

description of : [4.2.2. Getting Help](#)

evaluating : [13.3.3.1. The test command](#)

help for : [4.2.2. Getting Help](#)

processing : [13.3.1. Processing Arguments](#)

ASCII, non-ASCII characters : [5.3. Configuring X](#)

ash shell

[7.1.6. Configuring Access to Shells](#)

[13.1.1. A Variety of Shells](#)

asterisk (*) : [4.3.5.5. Finding a file](#)

authentication

password : [11.3. Using wvdial](#)

users

[11.3. Using wvdial](#)

[12.1. An FTP Server](#)

autoexec.bat file

renaming : [2.3.3.4. Using the fips utility.](#)

restoring name : [2.3.3.4. Using the fips utility.](#)

Axene Office : [8.1.4. Other Desktop Applications](#)

Index: B

back quote (`) : [13.2.9. Quoted Strings](#)

background properties (GNOME control center) : [6.7.3. The GNOME Control Center](#)

backslash (\) : [13.2.9. Quoted Strings](#)

backspace key : [5.3. Configuring X](#)

backups

[2.3.3.4. Using the fips utility](#)

[4.3.5.7. Working with compressed files](#)

[10.4.4. Using the Linux Samba Client for File Backup and Recovery](#)

before deleting files : [7.1.5. Deleting a User Account](#)

/etc/group file : [7.1.4.4. Removing a member from a group](#)

PC files : [10.3. Samba](#)

restoring : [10.4.4. Using the Linux Samba Client for File Backup and Recovery](#)

Samba server configuration : [10.3.6. Troubleshooting Samba](#)

backward slash (\) : [4.3.3.2. The directory tree](#)

bad blocks, scanning for : [3.1.2.10. Initializing and activating a swap partition](#)

base system

configuring : [3.1.2.15. Installing the base system](#)

installing : [3.1.2. Installing the Kernel and Base System](#)

base2_1.tgz file : [3.1.2.15. Installing the base system](#)

BASH shell

[7.1.1. Creating a User Account](#)

[13. Conquering the BASH Shell](#)

[13.1.1. A Variety of Shells](#)

assigned automatically : [7.1.6. Configuring Access to Shells](#)

automatic startup at login : [13.1.1. A Variety of Shells](#)

basics : [4.1.2. Logging In](#)

"Bourne Again SHell" : [13.1.1. A Variety of Shells](#)

built-in commands : [13.2.3. Filename Globbing](#)

commands for : [13.2.1. Typing Shell Commands](#)

conditional commands : [13.3.3. Conditional Logic](#)

learning, benefits of : [13.1.2. Why Learn to Use the Shell?](#)
Unix and : [13.1.2. Why Learn to Use the Shell?](#)
using : [13.2. Using the Shell](#)
Basic profile : [3.1.3.6. Choosing Packages](#)
Bell Telephone Lab : [1.2.2.1. The origins of Linux](#)
Berkeley System Distribution (BSD) : [1.2.2.1. The origins of Linux](#)
/bin file : [13.2.2. Commands and Arguments](#)
binary file : [12.1.1. Testing the FTP Server](#)
binary form : [1.2.2. How Linux is Different](#)
BIOS, system : [2.2.1. Information You Need](#)
Blizzard Entertainment : [9.2.3. Starcraft](#)
boot configuration : [4.1.1. Booting the System](#)
boot disk : [4.1.1. Booting the System](#)
 creating
 [2.3.3.4. Using the fips utility](#)
 [3.1.2.2. Booting from floppy diskettes](#)
 [3.1.2.17. Making a boot floppy](#)
boot prompt
 [3.1.2.3. Starting the installation procedure](#)
 [3.1.2.17. Making a boot floppy](#)
 [4.1.1. Booting the System](#)
booting
 Linux
 from CD-ROM with this book : [3.1.2. Installing the Kernel and Base System](#)
 from floppy disks : [3.1.2.2. Booting from floppy diskettes](#)
 from MS-DOS/Windows 95/98 : [3.1.2. Installing the Kernel and Base System](#)
 mailing list information : [3.2.4. The Debian Mailing Lists](#)
 system : [3.1.2.17. Making a boot floppy](#)
 your system
 [3.1.2.17. Making a boot floppy](#)
 [4.1.1. Booting the System](#)
"Bourne Again SHell" : (see [BASH shell](#))
Bourne shell : [7.1.6. Configuring Access to Shells](#)
brackets : [4.2.2.1. Using man](#)
broadcast address : [3.1.2.14. Configuring the network](#)

broadcast messages : [3.1.2.14. Configuring the network](#)

browseable files : [10.4.1. Microsoft Windows Client](#)

browseable shares : [10.4.3. Linux Client](#)

browsers : (see [web browsers](#))

BSD (Berkeley System Distribution) : [1.2.2.1. The origins of Linux](#)

buttons (seyon program) : [11.7.1. Using seyon](#)

Index: C

C programming language : [1.2.2.1. The origins of Linux](#)

C shell

[7.1.6. Configuring Access to Shells](#)

[13.1.1. A Variety of Shells](#)

caching documents : [12.2.1.3. The httpd.conf File](#)

Caldera OpenLinux : [1.2.2.6. Linux distributions](#)

Cancel button (GNOME control center) : [6.7.3. The GNOME Control Center](#)

cancel printing : [4.3.5.6. Printing a file](#)

canonical URL (web server) : [12.2.1.3. The httpd.conf File](#)

capplets (GNOME control center) : [6.7.3. The GNOME Control Center](#)

case condition : [13.3.3.3. The case command](#)

case sensitivity

[4.1.2. Logging In](#)

[4.2.1. Command Structure](#)

CD player (GNOME) : [6.6.2. GNOME CD Player](#)

CD-ROMs : [4.3.2. Filesystems](#)

CD-ROM drives : [2.1.3. Drives](#)

configuration information for

[2.2.1. Information You Need](#)

[2.2.2. Collecting Configuration Information by Using Windows](#)

module for, installing : [3.1.2.13. Configuring device driver modules](#)

CD-ROM with this book

base system : [3.1.2.15. Installing the base system](#)

booting Linux from : [3.1.2. Installing the Kernel and Base System](#)

FAQs : [3.2.2. FAQs](#)

gFTP client : [11.6. gFTP FTP client](#)

installation program : [3.1.2.12. Installing the Operating System Kernel and Modules](#)

packages, accessing : [3.1.3.5. Connecting via PPP](#)

central processing units : [2.1. Minimum Hardware Requirements](#)

cfdisk utility : [3.1.2.9. Partitioning a hard drive](#)

CGI directory : [12.2.1.2. The srm.conf File](#)

CGI scripts : [12.2.1.1. The access.conf file](#)

CHAP (Challenge-Handshake Authentication Protocol) : [11.3. Using wvdial](#)

character count : [13.2.6. Input/Output Redirection and Piping](#)
characters

(#) character : [13.2.1. Typing Shell Commands](#)

(&) character : [13.2.1. Typing Shell Commands](#)

(;) character : [13.2.1. Typing Shell Commands](#)

case condition : [13.3.3.3. The case command](#)

non-ASCII characters : [5.3. Configuring X](#)

special, for filename globbing : [13.2.3. Filename Globbing](#)

chat with Debian users : [3.2.6. Internet Relay Chat \(IRC\)](#)

child processes (web server) : [12.2.1.3. The httpd.conf File](#)

chkdsk utility : [2.3.3.4. Using the fips utility](#)

CLI (command line interface) : [4.1.3. Issuing Commands](#)

editing command line : [13.2.1. Typing Shell Commands](#)

clients

[10.4. Samba Client Configuration and Use](#)

[10.4.2. Other Clients](#)

SMB clients : [10.4.4. Using the Linux Samba Client for File Backup and Recovery](#)

clock settings : [5.3. Configuring X](#)

colon (:)

[7.1.4.4. Removing a member from a group](#)

[13.2.8. The Search Path](#)

colon prompt : [4.2.2.1. Using man](#)

color display

installation program : [3.1.2.4. Choosing color versus monochrome](#)

minicom program : [11.7.2.7. Running minicom](#)

combining files/directories : [4.3.5.7. Working with compressed files](#)

comma (,) : [7.1.4.4. Removing a member from a group](#)

mount options and : [7.2.1. Configuring Local Drives](#)

command line interface (CLI)

[4.1.3. Issuing Commands](#)

[13.2.1. Typing Shell Commands](#)

command line, adjusting to : [4.1.3. Issuing Commands](#)

command prompt : [4.1.2. Logging In](#)

MS-DOS : [1.2.3. Linux Features and Performance](#)

Unix : [1.2.2.5. The X Window System](#)

command-line arguments

installing modules and : [3.1.2.13. Configuring device driver modules](#)

commands

aliases for : [13.2.4. Shell Aliases](#)

for BASH shell : [13.2.1. Typing Shell Commands](#)

built-in : [13.2.3. Filename Globbing](#)

case condition : [13.3.3.3. The case command](#)

for condition : [13.3.3.6. The for command](#)

conditional commands : [13.3.3. Conditional Logic](#)

correcting : [4.1.4. Correcting Commands](#)

description of : [4.2.2. Getting Help](#)

for devices : [4.4. Working with Devices](#)

echo commands

[13.2.5. Shell Scripts](#)

[13.2.7. Shell Variables](#)

external : [13.2.2. Commands and Arguments](#)

for files : [4.3.5. Working with Files](#)

help for : [4.2.2. Getting Help](#)

if condition : [13.3.3.2. The if command](#)

issuing : [4. Issuing Linux Commands](#)

keystroke commands : [13.2.1. Typing Shell Commands](#)

for minicom program : [11.7.2.7. Running minicom](#)

prompting for confirmation : [13.2.4. Shell Aliases](#)

shell scripts and : [13.2.5. Shell Scripts](#)

structure of : [4.2.1. Command Structure](#)

subsystem commands : [12.1.1. Testing the FTP Server](#)

system administrator commands : [13.2.2. Commands and Arguments](#)

for viewing system status information : [4.5.1. Viewing System](#)

[Information](#)

while condition

[13.3.3.4. The while command](#)

[13.3.3.5. The until command](#)

comment token (#) : [12.2.1.3. The httpd.conf File](#)

Commit Changes button (Samba server) : [10.3.2.1. Configuring global variables](#)

Compress (PCTool) : [2.3.3.4. Using the fips utility](#)
compressed/compressing files : [4.3.5.7. Working with compressed files](#)
computers : (see [desktop computers](#))
conditional logic : [13.3.3. Conditional Logic](#)
Config item (dselect program) : [3.1.4. Installing the Applications](#)
config.sys file : [2.2.2. Collecting Configuration Information by Using Windows](#)
 renaming : [2.3.3.4. Using the fips utility](#)
 restoring name : [2.3.3.4. Using the fips utility](#)
configuration file (xf86config) : [5.3. Configuring X](#)
configuring
 collecting information for : [2.2. Collecting Information About Your System](#)
 configuration information : [2.2.1. Information You Need](#)
 information for exim mail package : [3.1.4.4. Installing the selected packages](#)
configuring base system : [3.1.2.15. Installing the base system](#)
configuring device driver modules : [3.1.2.13. Configuring device driver modules](#)
configuring dial-in shell server : [12.5. Configuring a Dial-In Shell Server](#)
configuring FTP server : [12.1. An FTP Server](#)
configuring GNOME desktop : [6.7. Configuring GNOME](#)
configuring group definitions : [7.1.4. Configuring Group Definitions](#)
configuring LAN (Local Area Network), Linux-based : [10.2. Network Administration](#)
configuring local drives : [7.2.1. Configuring Local Drives](#)
configuring mail server : [12.3. Using a Mail Server](#)
configuring minicom program : [11.7.2. Using minicom](#)
configuring modem : [11.2. Configuring Your Modem](#)
configuring networks : [3.1.2.14. Configuring the network](#)
 information needed for : [3.1.2.14. Configuring the network](#)
configuring packages : [C.4.6. Configuring Packages](#)
configuring Samba server : [10.3.2. Configuring Samba](#)
 file share parameters (Samba server) : [10.3.2.2. Configuring file share parameters](#)
configuring secure shell server : [12.4. Configuring a Secure Shell Server](#)
configuring swap partitions : [7.2.2. Configuring Swap Partitions](#)

configuring web server : [12.1.1. Testing the FTP Server](#)
configuring X Window System : [5.3. Configuring X](#)
caution with : [5.2. Installing X](#)
configuring your new system : [3.1.3. Configuring the New System](#)
confirmation screen : [3.1.2.14. Configuring the network](#)
connecting to the Internet
via PPP : [3.1.3.5. Connecting via PPP](#)
with seyon program : [11.7.1. Using seyon](#)
connections, persistent (web server) : [12.2.1.3. The httpd.conf File](#)
control center (GNOME desktop) : [6.7.3. The GNOME Control Center](#)
control characters : [4.5.2. Using the ae Editor](#)
control keystrokes : [13.2.1. Typing Shell Commands](#)
Control Panel, Windows 95/98 : [2.2.2. Collecting Configuration Information by Using Windows](#)
copying
files
[4.3.5.3. Copying a file](#)
[6.5.4. Using the File Manager](#)
text : [6.2.1. Copying and Pasting Text](#)
copyright/copyleft : [1.2.2.3. Copyleft](#)
CORBA (Common Object Request Broker Architecture) : [6.3.2. Enlightenment](#)
Corel WordPerfect for Linux : (see [WordPerfect for Linux](#))
CPUs (Central Processing Units) : [2.1. Minimum Hardware Requirements](#)
Create Launcher Applet dialog box : [6.7.1. The GNOME Panel](#)
cron program : [13.3.6. Continuing Onward](#)
csh shell : [7.1.6. Configuring Access to Shells](#)
CTS (modem indicator) : [11.7.1. Using seyon](#)
curly braces ({}): [13.2.7. Shell Variables](#)
current working directory : [4.3.3.1. Home and working directories](#)
changing : [13.2.7. Shell Variables](#)
cycle of system use : [4.1. The System Use Cycle](#)
cylinders : [2.3.1. How Hard Disks are Organized](#)
cylinder for new partition : [2.3.3.4. Using the fips utility](#)
Cyrix processors : [2.1.1. Central Processing Unit \(CPU\)](#)

Index: D

daemons

[C.6.5.2. Installing the samba package](#)

[7.3.2. Starting and Stopping Services](#)

[10.3.3. Viewing Samba Server Status](#)

httpd daemon

[12.2.1. Configuration](#)

[12.2.1.3. The httpd.conf File](#)

PPP daemon : [11.1. Connecting to the Internet](#)

data storage : [1.2.3. Linux Features and Performance](#)

data streams, standard : [13.2.6. Input/Output Redirection and Piping](#)

database

for packages : [C.3.3. Querying the Package Database](#)

relational : [8.1.4. Other Desktop Applications](#)

selecting video card from : [5.3. Configuring X](#)

date

applet for (GNOME desktop) : [6.5.3.5. Date & time applet](#)

displaying : [4.1.3. Issuing Commands](#)

setting/storing : [7.5. Setting the System Time and Time Zone](#)

deb : [C.1. Packages](#)

Debian GNU/Linux

(see also [entries at Linux](#))

[1.2.2.6. Linux distributions](#)

[1.4.1. Web Pages](#)

FAQ (Frequently Asked Questions) : [3.2.2. FAQs](#)

FAQs (Frequently Asked Questions) : [1.4.1. Web Pages](#)

help : [3.2. Getting Help](#)

install utility : [2.3.3.2. Identifying an unused partition](#)

mailing lists : [3.2.4. The Debian Mailing Lists](#)

using Linux applications and clients with : [C.6.3. Chapter 8: Using Linux Applications and Clients](#)

web site : [3.2.1. The Debian Web Site](#)

DEC Unix : [1.1. Linux at Home and at Work](#)

defaults

- aliases, viewing : [13.2.4. Shell Aliases](#)
- configuration : [4.1.1. Booting the System](#)
- font : [6.6.1. GNOME Terminal](#)
- network route, deleting : [13.3.4. Periscope: A Useful Networking Script](#)
- text editor (GNOME control center) : [6.7.3. The GNOME Control Center](#)
- defrag utility (Microsoft) : [2.3.3.4. Using the fips utility](#)
- defragmenting hard disks : [2.3.3.4. Using the fips utility](#)
- deleting
 - directories : [4.3.4.5. Removing a directory](#)
 - files
 - [4.3.5.2. Removing a file](#)
 - [6.5.4. Using the File Manager](#)
 - home directory : [7.1.5. Deleting a User Account](#)
 - packages
 - pre-selection : [C.4.3. Choosing Packages for Installation or Removal](#)
 - partitions : [2.3.3.2. Identifying an unused partition](#)
 - user accounts : [7.1.5. Deleting a User Account](#)
- dependency conflicts : [C.4.3. Choosing Packages for Installation or Removal](#)
- description strings
 - for monitors : [5.3. Configuring X](#)
 - for video cards : [5.3. Configuring X](#)
- desktop (GNOME desktop) : [6.5.3.2. Desktop](#)
- desktop applications
 - available for Linux : [8.1.4. Other Desktop Applications](#)
 - desktop publishing applications : [8.1.4. Other Desktop Applications](#)
 - desktop suites
 - StarOffice (Star Division) : [8.1.2. StarOffice](#)
 - WordPerfect for Linux (Corel) : [8.1.3. Corel WordPerfect for Linux](#)
 - other : [8.1.4. Other Desktop Applications](#)
- desktop computers
 - Linux run on : [1.2.1. PC Operating Systems](#)
- operating systems for

[1.2.1. PC Operating Systems](#)

[1.2.3. Linux Features and Performance](#)

sales of : [1.2.1. PC Operating Systems](#)

using Linux : [1.2.2.4. The Linux kernel](#)

desktop environments : [6. Using the X Window System](#)

desktop publishing applications : [8.1.4. Other Desktop Applications](#)

desktop suites : [8.1.4. Other Desktop Applications](#)

StarOffice : [8.1.2. StarOffice](#)

StarOffice (Star Division) : [8.1.2. StarOffice](#)

WordPerfect for Linux (Corel) : [8.1.3. Corel WordPerfect for Linux](#)

desktop themes (GNOME control center) : [6.7.3. The GNOME Control Center](#)

desktops

[6.4. Desktops](#)

[6.5.1. Launching GNOME and Enlightenment](#)

developers : [1.2.2. How Linux is Different](#)

device driver modules, configuring : [3.1.2.13. Configuring device driver modules](#)

device files : [5.3. Configuring X](#)

Device Manager, Windows 95/98 : [2.2.2. Collecting Configuration Information by Using Windows](#)

devices

commands for : [4.4. Working with Devices](#)

list of : [4.3.1. Devices](#)

troubleshooting : [3.1.2.13. Configuring device driver modules](#)

DIAGNOSIS.txt file : [10.3.6. Troubleshooting Samba](#)

dial-in shell server, configuring : [12.5. Configuring a Dial-In Shell Server](#)

Dial-Up Networking (Microsoft Windows) : [11.1. Connecting to the Internet](#)

Dialing Directory dialog box

[11.7.2.7. Running minicom](#)

[11.7.1. Using seyon](#)

dialing options, configuring (minicom program) : [11.7.2.4. Configuring modem and dialing options](#)

dialogs : [11.3. Using wvdial](#)

Digital Equipment Corporation : [5.1. What is X?](#)

directories

[4.3.2. Filesystems](#)

[4.3.3. Directories](#)

combining : [4.3.5.7. Working with compressed files](#)

contents of : [4.3.4.3. Displaying directory contents](#)

creating : [4.3.4.4. Creating a directory](#)

new search directory : [13.2.8. The Search Path](#)

deleting : [4.3.4.5. Removing a directory](#)

My Documents directory : [4.3.4.4. Creating a directory](#)

naming : [4.3.4.4. Creating a directory](#)

removing : [4.3.4.5. Removing a directory](#)

renaming : [4.3.5.4. Renaming or moving a file](#)

directory tree : [4.3.3.2. The directory tree](#)

disabling virtual memory : [2.3.3.4. Using the fips utility](#)

Disk Drives icon, Windows 95/98 : [2.2.2. Collecting Configuration Information by Using Windows](#)

disk drives, fixed : [2.3.2. Viewing Partition Information](#)

disk management : [1.2.3. Linux Features and Performance](#)

disks, hard : (see [hard disks](#))

displaying system status : [4.1.3. Issuing Commands](#)

displays : (see [monitors](#))

distribution : (see [Linux distributions](#); [software distribution](#))

DNS (Domain Name Server) : [10.1. Introduction](#)

documenting your system configuration : [2.2. Collecting Information About Your System](#)

documents

 caching (web server) : [12.2.1.3. The httpd.conf File](#)

 format and organization of (web server) : [12.2.1.2. The srm.conf File](#)

 permissions (web server) : [12.2.1.1. The access.conf file](#)

dollar sign (\$)

[3.1.4.5. Exiting dselect](#)

[13.2.7. Shell Variables](#)

domain name

[3.1.2.14. Configuring the network](#)

[12.2.2. Startup and Use](#)

Domain Name Server (DNS) : [10.1. Introduction](#)

Doom : [9.2.1. DOOM](#)

 installing : [C.6.4.1. Installing Doom](#)

DOSKEYS editor (MS_DOS) : [13.2.1. Typing Shell Commands](#)
\dosutils\fips20\errors.txt file : [2.3.3.4. Using the fips utility](#)
\dosutils\fips20\fips.exe file : [2.3.3.4. Using the fips utility](#)
\dosutils\fips20\restorrb.exe file : [2.3.3.4. Using the fips utility](#)
dot (.)

[3.1.2.14. Configuring the network](#)

[4.3.4.3. Displaying directory contents](#)

in filename : [4.3.4.4. Creating a directory](#)

dot character (.) : [7.1.1. Creating a User Account](#)

double quote inxs (") : [13.2.9. Quoted Strings](#)

downloads

[11.7.2.1. Configuring filenames and paths](#)

[11.7.2.2. Configuring file transfer protocols](#)

dpkg package management tool : [C.3. Using dpkg](#)

drag and drop operations : [6.5.4. Using the File Manager](#)

drawers (GNOME desktop) : [6.7.1. The GNOME Panel](#)

drawing program (StarDraw) : [8.1.2.2. Running StarOffice](#)

drive icon (GNOME desktop) : [6.5.3.3. Drive icon](#)

drive letters

[2.3.1. How Hard Disks are Organized](#)

[2.3.3.2. Identifying an unused partition](#)

[2.3.3.4. Using the fips utility](#)

mapping to file share : [10.4.1. Microsoft Windows Client](#)

drives

hard : (see [hard disks](#))

local, configuring : [7.2.1. Configuring Local Drives](#)

dselect package management tool : [C.4. Using dselect](#)

dselect program : [3.1.3.6. Choosing Packages](#)

exiting : [3.1.4.5. Exiting dselect](#)

DSR (modem indicator) : [11.7.1. Using seyon](#)

DTR (modem indicator) : [11.7.1. Using seyon](#)

dual boot : [1.3. Reasons to Choose or Not Choose Linux](#)

Index: E

echo commands

[13.2.5. Shell Scripts](#)

[13.2.7. Shell Variables](#)

editing command line : [13.2.1. Typing Shell Commands](#)

ELF (Extension Language Facility) : [8.1.1.1. Running Applixware](#)

email

fetchmail package for receiving/reading : [12.3. Using a Mail Server](#)

PGP-encrypted : [8.1.2.2. Running StarOffice](#)

sending/mail server for sending/receiving : [12.3. Using a Mail Server](#)

of server administrator : [12.2.1.3. The httpd.conf File](#)

travelling and : [13.3.4. Periscope: A Useful Networking Script](#)

enabling virtual memory : [2.3.3.4. Using the fips utility](#)

encryption : [12.4. Configuring a Secure Shell Server](#)

Enlightenment window manager

[6.2.1. Copying and Pasting Text](#)

[6.3.2. Enlightenment](#)

configuring : [6.5.1. Launching GNOME and Enlightenment](#)

using : [6.5.1. Launching GNOME and Enlightenment](#)

environment variables : (see [shells, variables](#) for)

error files, web server : [12.2.1.3. The httpd.conf File](#)

error messages : [13.2.6. Input/Output Redirection and Piping](#)

file recorded in : [7.4. Viewing System Messages and Logs](#)

error stream, standard : [13.2.6. Input/Output Redirection and Piping](#)

/etc/adduser.conf file : [7.1.6. Configuring Access to Shells](#)

/etc/fstab file : [7.2.1. Configuring Local Drives](#)

/etc/fstab file : [4.4.1. Mounting and Unmounting Drives](#)

/etc/group file

[7.1.4.1. Creating a group](#)

[7.1.4.4. Removing a member from a group](#)

/etc/httpd/conf file : [12.2.1. Configuration](#)

/etc/inetd.conf file : [C.6.5.2. Installing the samba package](#)

/etc/passwd file : [7.1.3. Changing a User Account Password](#)

/etc/smb.conf file : [10.3.2. Configuring Samba](#)

/etc/shadow file : [7.1.3. Changing a User Account Password](#)
/etc/shells file : [7.1.6. Configuring Access to Shells](#)
/etc/smb.conf file : [10.3.1. Samba Server Installation](#)
/etc/smbusers configuration file : [10.3.1. Samba Server Installation](#)
(eth0) Ethernet interface : [13.3.4. Periscope: A Useful Networking Script](#)
Ethernet : [3.1.2.14. Configuring the network](#)
Ethernet interface (eth0) : [13.3.4. Periscope: A Useful Networking Script](#)
/etc/smb.conf file : [10.3.4. Viewing Samba Server Configuration](#)
Excel (Microsoft) : [8.1.1.1. Running Applixware](#)
exim mail package : [3.1.4.4. Installing the selected packages](#)
 configuring after installation : [3.1.4.4. Installing the selected packages](#)
 log files for : [7.4. Viewing System Messages and Logs](#)
exit codes : [13.3.2. Exit Codes](#)
exiting X Window System
 [5.4. Starting and Stopping X](#)
 [6.1.3. Terminating X](#)
exiting your system : [4.1.6. Logging Out](#)
expanding files : [4.3.5.7. Working with compressed files](#)
EXT2 filesystem : [1.2.3. Linux Features and Performance](#)
extended partitions : [3.1.2.9. Partitioning a hard drive](#)
Extension Language Facility (ELF) : [8.1.1.1. Running Applixware](#)
external commands : [13.2.2. Commands and Arguments](#)
 search path and : [13.2.8. The Search Path](#)

Index: F

FAQ (Frequently Asked Questions) : [1.4.2. Newsgroups](#)

FAQs (Frequently Asked Questions)

[1.4.1. Web Pages](#)

[3.2.2. FAQs](#)

FAT filesystems

[1.2.3. Linux Features and Performance](#)

[2.3.1. How Hard Disks are Organized](#)

FAT partitions : [2.3.3.4. Using the fips utility](#)

fdisk utility

[2.3.2. Viewing Partition Information](#)

[2.3.3.2. Identifying an unused partition](#)

features of Linux : [1.2.3. Linux Features and Performance](#)

fetchmail package : [12.3. Using a Mail Server](#)

file manager

[6.5.3.1. Home directory icon](#)

[6.5.3.10. File Manager](#)

file permissions : [4.3.5.9. Working with file permissions](#)

file recovery : [10.4.4. Using the Linux Samba Client for File Backup and Recovery](#)

file shares

active, viewing : [10.3.3. Viewing Samba Server Status](#)

mapping to drive letter : [10.4.1. Microsoft Windows Client](#)

options for (Samba server) : [10.3.2.2. Configuring file share parameters](#)

parameters for (Samba server) : [10.3.2.2. Configuring file share parameters](#)

file sharing : [10.3. Samba](#)

File Transfer Protocol : (see [FTP](#))

file transfer protocols, configuring (minicom program) : [11.7.2.2. Configuring file transfer protocols](#)

filename completion : [13.2.10. The Power of the Linux Shell](#)

filename globbing : [13.2.3. Filename Globbing](#)

filename metacharacters : [13.2.3. Filename Globbing](#)

filenames

configuring (minicom program) : [11.7.2.1. Configuring filenames and paths](#)

containing dot (.)

[4.3.4.3. Displaying directory contents](#)

[4.3.4.4. Creating a directory](#)

globbing of : [13.2.3. Filename Globbing](#)

files

[4.3.2. Filesystems](#)

[4.3.4.5. Removing a directory](#)

backing up before deleting : [7.1.5. Deleting a User Account](#)

binary : [12.1.1. Testing the FTP Server](#)

character count : [13.2.6. Input/Output Redirection and Piping](#)

combining : [4.3.5.7. Working with compressed files](#)

commands for : [4.3.5. Working with Files](#)

compressed/compressing : [4.3.5.7. Working with compressed files](#)

copying

[4.3.5.3. Copying a file](#)

[6.5.4. Using the File Manager](#)

deleting

[4.3.5.2. Removing a file](#)

[6.5.4. Using the File Manager](#)

displaying contents of : [4.3.5.1. Displaying the contents of a file](#)

expanding : [4.3.5.7. Working with compressed files](#)

finding : [4.3.5.5. Finding a file](#)

hidden

[2.3.3.4. Using the fips utility](#)

[4.3.4.3. Displaying directory contents](#)

[13.2.5. Shell Scripts](#)

installed from package, listing : [C.3.3.4. List files installed from a package](#)

line count : [13.2.6. Input/Output Redirection and Piping](#)

links and : [4.3.5.8. Working with links](#)

list of, displaying : [7.1.1. Creating a User Account](#)

moving

[4.3.5.4. Renaming or moving a file](#)

[6.5.4. Using the File Manager](#)

multimedia : [12.2.1.2. The srm.conf File](#)
null : [13.2.6. Input/Output Redirection and Piping](#)
old/new : [4.3.5.4. Renaming or moving a file](#)
open, viewing : [10.3.3. Viewing Samba Server Status](#)
permissions : [4.3.5.9. Working with file permissions](#)
print job number : [4.3.5.6. Printing a file](#)
printing : [4.3.5.6. Printing a file](#)
queueing to printer : [4.3.5.6. Printing a file](#)
redirecting input/output to : [13.2.6. Input/Output Redirection and Piping](#)

renaming

[4.3.5.4. Renaming or moving a file](#)

[6.5.4. Using the File Manager](#)

with links : [4.3.5.8. Working with links](#)

tar files : [4.3.5.7. Working with compressed files](#)

transferring via FTP : [12.1.1. Testing the FTP Server](#)

uncompressing : [4.3.5.7. Working with compressed files](#)

uploads

[11.6. gFTP FTP client](#)

[11.7.2.1. Configuring filenames and paths](#)

[11.7.2.2. Configuring file transfer protocols](#)

wordcount : [13.2.6. Input/Output Redirection and Piping](#)

filesystems

[1.2.3. Linux Features and Performance](#)

[2.3.1. How Hard Disks are Organized](#)

[4.3.2. Filesystems](#)

administering : [7.2. Filesystem Administration](#)

CD-ROM filesystem : [7.2.1. Configuring Local Drives](#)

mounting

[7.2. Filesystem Administration](#)

[7.2.1. Configuring Local Drives](#)

MS-DOS filesystem : [7.2.1. Configuring Local Drives](#)

types of : [4.3.2. Filesystems](#)

Windows 95/98 : [4.3.4.4. Creating a directory](#)

filters : [13.2.6. Input/Output Redirection and Piping](#)

finding files : [4.3.5.5. Finding a file](#)

finding Linux : (see [Linux distributions](#))

fips utility : [2.3.3.4. Using the fips utility](#)

floppy disks : [4.3.2. Filesystems](#)

formatting : [4.4.2. Formatting Floppy Diskettes](#)

mounted by users : [7.2.1. Configuring Local Drives](#)

mounting : [4.4.2. Formatting Floppy Diskettes](#)

unmounting : [4.4.2. Formatting Floppy Diskettes](#)

floppy drives : [2.1.3. Drives](#)

designators : [4.4.2. Formatting Floppy Diskettes](#)

for condition : [13.3.3.6. The for command](#)

format names, web server : [12.2.1.3. The httpd.conf File](#)

formatting partitions : [2.3.1. How Hard Disks are Organized](#)

forward slash (/)

[4.3.3.2. The directory tree](#)

[4.3.4.4. Creating a directory](#)

forwarding address : [12.2.1.2. The srm.conf File](#)

free software : [1.2.2.2. Free software](#)

Free Software Foundation (FSF) : [1.2.2.2. Free software](#)

FTP (File Transfer Protocol)

client for Linux : [11.6. gFTP FTP client](#)

FTP (File Transfer Protocol)

anonymous : [12.1. An FTP Server](#)

hackers and : [12.1. An FTP Server](#)

subsystem prompt : [12.1.1. Testing the FTP Server](#)

FTP client (Linux) : [12.1.1. Testing the FTP Server](#)

FTP client (Microsoft Windows) : [12.1.1. Testing the FTP Server](#)

FTP server

installing/configuring : [12.1. An FTP Server](#)

newsgroups for : [12.1.1. Testing the FTP Server](#)

password : [12.1.1. Testing the FTP Server](#)

subsystem commands for : [12.1.1. Testing the FTP Server](#)

testing : [12.1.1. Testing the FTP Server](#)

userid : [12.1.1. Testing the FTP Server](#)

FTP sites : [2.1.3. Drives](#)

Full View button (swat tool) : [10.3.4. Viewing Samba Server Configuration](#)

FVWM window manager : [6.3.1. FVWM](#)

Index: G

games : [9. Playing Linux Games](#)

Doom : [9.2.1. DOOM](#)

GNOME : [6.4.2. GNOME](#)

available with KDE desktop : [6.4.1. KDE](#)

popular games : [9.2. Closeups of Some Popular Games](#)

Quake II : [9.2.2. Quake II](#)

Starcraft : [9.2.3. Starcraft](#)

gateways

[3.1.2.14. Configuring the network](#)

[10.1. Introduction](#)

General Electric : [1.2.2.1. The origins of Linux](#)

geometry of hard disks : [2.3.1. How Hard Disks are Organized](#)

gFTP client : [11.6. gFTP FTP client](#)

global variables (Samba server)

configuring : [10.3.2.1. Configuring global variables](#)

options for : [10.3.2.1. Configuring global variables](#)

Globals button (Samba server) : [10.3.2.1. Configuring global variables](#)

GNOME : [C.6.2. Chapter 6: Using X Windows](#)

GNOME CD player : [6.6.2. GNOME CD Player](#)

GNOME desktop

[6.3.1. FVWM](#)

[6.3.2. Enlightenment](#)

[6.4.2. GNOME](#)

applets and applications : [6.6. Using GNOME Applets and Applications](#)

configuring : [6.7. Configuring GNOME](#)

logging out : [6.5.2. Logging Out](#)

using : [6.5.1. Launching GNOME and Enlightenment](#)

GNOME terminal : [6.6.1. GNOME Terminal](#)

GNOME-compliant applications, appearance of : [6.7.3. The GNOME Control Center](#)

GNU GPL : (see [GNU Public License](#))

GNU Public License

[1.2.2.3. Copyleft](#)

[1.2.2.6. Linux distributions](#)

[1.2.3. Linux Features and Performance](#)

GNU's not Unix : [1.2.2.2. Free software](#)

graphical applications

available with KDE desktop : [6.4.1. KDE](#)

graphical user interface

(see also [X Window System](#))

[1.2.2.5. The X Window System](#)

[13.1.2. Why Learn to Use the Shell?](#)

newsgroups discussion of : [13.1.2. Why Learn to Use the Shell?](#)

graphics adaptors : (see [video cards](#))

graphics applications : [1.1. Linux at Home and at Work](#)

group access permission : [4.3.5.9. Working with file permissions](#)

group definitions, configuring : [7.1.4. Configuring Group Definitions](#)

Group ID : [7.1.4.4. Removing a member from a group](#)

groups

[4.3.4.3. Displaying directory contents](#)

[4.3.5.9. Working with file permissions](#)

administering : [7.1. User and Group Administration](#)

creating : [7.1.4.1. Creating a group](#)

deleting : [7.1.4.2. Deleting a group](#)

members

adding : [7.1.4.3. Adding a member to a group](#)

removing : [7.1.4.4. Removing a member from a group](#)

web server : [12.2.1.3. The httpd.conf File](#)

guest operating systems : [8.2.1. The VMware Virtual Platform](#)

.gz files : [4.3.5.7. Working with compressed files](#)

Index: H

hackers

- encrypted passwords and : [3.1.3.3. Configuring shadow passwords](#)

- FTP and : [12.1. An FTP Server](#)

- secure shell servers and : [12.4. Configuring a Secure Shell Server](#)

- security options and : [10.3.2.1. Configuring global variables](#)

hanging up (seyon program) : [11.7.1. Using seyon](#)

hard disk drives : (see [hard disks](#))

hard disk space : [2.1.3. Drives](#)

- obtaining sufficient : [2.3.3. Obtaining Sufficient Disk Space](#)

hard disks : [2.3.2. Viewing Partition Information](#)

- booting from : [4.1.1. Booting the System](#)

- checking condition of : [2.3.3.4. Using the fips utility](#)

- configuration information for : [2.2.2. Collecting Configuration](#)

- [Information by Using Windows](#)

- defragmenting : [2.3.3.4. Using the fips utility](#)

- installing new : [2.3.3.1. Installing a new disk drive](#)

- organization of : [2.3.1. How Hard Disks are Organized](#)

- partitioning : [3.1.2.9. Partitioning a hard drive](#)

- preparing for installation : [2.3. Preparing Your Hard Disk](#)

- selecting : [3.1.2.8. Selecting the hard drive](#)

- structure of : [2.3.1. How Hard Disks are Organized](#)

- viewing partition information : [2.3.2. Viewing Partition Information](#)

hard drives : (see [hard disks](#))

hard links : [4.3.5.8. Working with links](#)

hardware requirements : [2.1. Minimum Hardware Requirements](#)

hash mark (#) : [7.2.1. Configuring Local Drives](#)

header lines : [4.2.1. Command Structure](#)

heads : [2.3.1. How Hard Disks are Organized](#)

help : [3.2. Getting Help](#)

- for arguments : [4.2.2. Getting Help](#)

- for commands : [4.2.2. Getting Help](#)

- on CDs : [6.6.2. GNOME CD Player](#)

- for options : [4.2.2. Getting Help](#)

help viewer (GNOME desktop) : [6.5.3.9. Help viewer](#)

Hess, Joey : [C.1. Packages](#)

hidden files

[2.3.3.4. Using the fips utility](#)

[4.3.4.3. Displaying directory contents](#)

[7.1.1. Creating a User Account](#)

[13.2.5. Shell Scripts](#)

hide button (GNOME desktop) : [6.5.3.8. Hide button](#)

history list

[4.1.4. Correcting Commands](#)

[13.2.1. Typing Shell Commands](#)

[13.2.10. The Power of the Linux Shell](#)

home directory : [4.3.3.1. Home and working directories](#)

deleting : [7.1.5. Deleting a User Account](#)

for new accounts : [7.1.1. Creating a User Account](#)

home directory icon (GNOME desktop) : [6.5.3.1. Home directory icon](#)

horizontal sync rate (xf86config) : [5.3. Configuring X](#)

caution with : [5.3. Configuring X](#)

Host Information dialog box : [10.2.6. Miscellaneous Network Configuration Options](#)

host operating systems : [8.2.1. The VMware Virtual Platform](#)

hostnames

[10.1. Introduction](#)

[12.2.2. Startup and Use](#)

choosing : [3.1.2.14. Configuring the network](#)

hosts file : [10.2.6. Miscellaneous Network Configuration Options](#)

search path : [10.2.5. Hostname Search Path](#)

web server : [12.2.1.3. The httpd.conf File](#)

hosts allow option (Samba server) : [10.3.2.1. Configuring global variables](#)

hosts, permissions for : [12.2.1.1. The access.conf file](#)

HOWTOs : [3.2.3. HOWTOs](#)

.htaccess file : [12.2.1.1. The access.conf file](#)

httpd daemon

[12.2.1. Configuration](#)

[12.2.1.3. The httpd.conf File](#)

httpd.conf file : [12.2.1.3. The httpd.conf File](#)

Hyperterminal (Microsoft Windows)

[11.1. Connecting to the Internet](#)

[12.5.1. Configuration](#)

Index: I

ibmbio.com/ibmdos.com files : [2.3.3.4. Using the fips utility](#).

icons : [12.2.1.2. The srm.conf File](#)

id Software

[C.6.4.1. Installing Doom](#)

[9.2.1. DOOM](#)

[9.2.2. Quake II](#)

identification strings

for monitors : [5.3. Configuring X](#)

for video cards : [5.3. Configuring X](#)

if condition : [13.3.3.2. The if command](#)

IMAGE : [2.3.3.4. Using the fips utility](#).

indexes : [12.2.1.2. The srm.conf File](#)

inetd.conf file : [12.1.1. Testing the FTP Server](#)

information

gathering about your system : [2.2. Collecting Information About Your System](#)

gathering via Microsoft Windows : [2.2.2. Collecting Configuration Information by Using Windows](#)

init process : [12.5.1. Configuration](#)

initializing swap partitions : [3.1.2.10. Initializing and activating a swap partition](#)

inittab file : [12.5.1. Configuration](#)

input data : [13.2.6. Input/Output Redirection and Piping](#)

input focus : [6.1. Keyboard Operations](#)

input stream, standard : [13.2.6. Input/Output Redirection and Piping](#)

input/output redirectors : [13.2.6. Input/Output Redirection and Piping](#)

insecure passwords : [3.1.3.1. Establishing a root password](#)

installing base system

[3.1.2. Installing the Kernel and Base System](#)

[3.1.2.15. Installing the base system](#)

installing FTP server : [12.1. An FTP Server](#)

installing kernel

[3.1.2. Installing the Kernel and Base System](#)

[3.1.2.12. Installing the Operating System Kernel and Modules](#)
installing Linux
 [2. Preparing to Install Linux](#)
 [3. Installing Linux](#)
 before installing Linux : [3.1. Installing the Operating System and Applications](#)
 Debian GNU/Linux install utility : [2.3.3.2. Identifying an unused partition](#)
 hardware requirements : [2.1. Minimum Hardware Requirements](#)
 installation program : [3.1.1. The Installation Program User Interface](#)
 main menu : [3.1.2.6. The Installation Main Menu](#)
 selecting color/monochrome display : [3.1.2.4. Choosing color versus monochrome](#)
 phases of : [3.1. Installing the Operating System and Applications](#)
 preparation for : [2. Preparing to Install Linux](#)
 starting the process : [3.1.2.3. Starting the installation procedure](#)
installing modules
 [3.1.2.12. Installing the Operating System Kernel and Modules](#)
 [3.1.2.13. Configuring device driver modules](#)
installing new disk drives : [2.3.3.1. Installing a new disk drive](#)
installing packages : [C. The Debian Package Management Utilities](#)
 apt-get management tool : [C.5.2.2. Installing a Package](#)
 dpkg management tool : [C.3.1. Installing a Package](#)
 dselect management tool : [C.4.5. Installing Packages](#)
 pre-selection : [C.4.3. Choosing Packages for Installation or Removal](#)
installing Samba server : [10.3.1. Samba Server Installation](#)
installing StarOffice : [8.1.2.1. Installing StarOffice](#)
installing the X Window System : [5.2. Installing X](#)
installing web server : [12.1.1. Testing the FTP Server](#)
installing WordPerfect for Linux (Corel) : [8.1.3.1. Installing WordPerfect for Linux](#)
Intel processors : [2.1.1. Central Processing Unit \(CPU\)](#)
Intel-architecture PCs : [1.2.3. Linux Features and Performance](#)
interfaces
 [1.2.2.5. The X Window System](#)
 [10.1. Introduction](#)
internal server process date (web server) : [12.2.1.3. The httpd.conf File](#)

Internet : [1.3. Reasons to Choose or Not Choose Linux](#)
connecting to : [11. Getting Connected to the Internet](#)
service provider (ISP)

[11.1. Connecting to the Internet](#)

[13.3.4. Periscope: A Useful Networking Script](#)

Internet name : (see [hostnames](#))

Internet Relay Chat IRC) : [3.2.6. Internet Relay Chat \(IRC\)](#).

IP address

[3.1.2.14. Configuring the network](#)

[13.3.5. Using periscope](#)

finding for your computer : [13.3.4. Periscope: A Useful Networking Script](#)

hostname search path : [10.2.5. Hostname Search Path](#)

hosts file : [10.2.6. Miscellaneous Network Configuration Options](#)

specifying for LAN (Local Area Network) : [10.2.5. Hostname Search Path](#)

web server : [12.2.1.3. The httpd.conf File](#)

IP number : [10.1. Introduction](#)

IRC (Internet Relay Chat) : [3.2.6. Internet Relay Chat \(IRC\)](#).

ISP (Internet service provider)

[11.1. Connecting to the Internet](#)

[13.3.4. Periscope: A Useful Networking Script](#)

issuing commands : [4. Issuing Linux Commands](#)

from the command line : [4.1.3. Issuing Commands](#)

Index: J

Joy, Bill : [1.2.2.1. The origins of Linux](#)

Index: K

K Desktop Environment : [6.4.1. KDE](#)

KDE desktop : [6.4.1. KDE](#)

kermit program : [11.7.2.1. Configuring filenames and paths](#)

kernels : [1.2.2.4. The Linux kernel](#)

installing

[3.1.2. Installing the Kernel and Base System](#)

[3.1.2.12. Installing the Operating System Kernel and Modules](#)

options : [3.1.2.3. Starting the installation procedure](#)

keyboard

bell and sounds (GNOME control center) : [6.7.3. The GNOME Control Center](#)

configuring options for (minicom program) : [11.7.2.5. Configuring screen and keyboard options](#)

extended : [5.3. Configuring X](#)

keyboard map : [5.3. Configuring X](#)

operations with : [6.1. Keyboard Operations](#)

options for, configuring (minicom program) : [11.7.2.5. Configuring screen and keyboard options](#)

properties (GNOME control center) : [6.7.3. The GNOME Control Center](#)

selecting : [3.1.2.7. Selecting a keyboard](#)

keystrokes

caution with : [3.1.1.2. Common keystrokes](#)

commands for : [13.2.1. Typing Shell Commands](#)

control keystrokes : [13.2.1. Typing Shell Commands](#)

for editing : [13.2.1. Typing Shell Commands](#)

for installation program : [3.1.1.2. Common keystrokes](#)

screen savers, restoring : [4.1.5. Using Virtual Consoles](#)

Tab keystroke : [13.2.1. Typing Shell Commands](#)

for virtual consoles : [3.1.1.3. Using virtual consoles](#)

virtual consoles, switching between : [4.1.5. Using Virtual Consoles](#)

keywords, searching by : [4.2.2.2. Using apropos](#)

KOffice

[6.4.1. KDE](#)

[8.1.4. Other Desktop Applications](#)

Korn shell

[7.1.6. Configuring Access to Shells](#)

[13.1.1. A Variety of Shells](#)

ksh shell : [7.1.6. Configuring Access to Shells](#)

KWM (K Window Manager) : [6.4.1. KDE](#)

Index: L

LAN (Local Area Network), Linux-based

administering : [10.2. Network Administration](#)

setting up

[C.6.5.1. Installing the lynx package](#)

[10. Setting Up a Linux-Based LAN](#)

LanManager : [10.3. Samba](#)

laptop computers

Debian GNU/Linux and : [3.2.4. The Debian Mailing Lists](#)

large disk support : [2.3.2. Viewing Partition Information](#)

launchers : [6.7.1. The GNOME Panel](#)

launching programs : [13.2.2. Commands and Arguments](#)

search path and : [13.2.8. The Search Path](#)

LBA (logical block addressing) : [2.3.1. How Hard Disks are Organized](#)

left mouse button : [6.2.2. Using Scrollbars](#)

legacy applications : [8.2.1. The VMware Virtual Platform](#)

licenses

[1.2.1. PC Operating Systems](#)

[1.2.2.3. Copyleft](#)

QT : [6.4.1. KDE](#)

line count : [13.2.6. Input/Output Redirection and Piping](#)

links

[4.3.4.3. Displaying directory contents](#)

[4.3.5.8. Working with links](#)

Linux

(see also [entries at Debian GNU/Linux](#))

[1. Why Run Linux?](#)

[1.2.3. Linux Features and Performance](#)

administering : [7. Configuring and Administering Linux](#)

BASH shell : [13. Conquering the BASH Shell](#)

case sensitivity : [4.1.2. Logging In](#)

command interpreter : [13.1. The Linux Shell](#)

commands : [4. Issuing Linux Commands](#)

compared : [1.4.2. Newsgroups](#)

configuring : [7. Configuring and Administering Linux](#)
distributions

[1.2.2. How Linux is Different](#)

[1.2.2.6. Linux distributions](#)

features : [1.2.3. Linux Features and Performance](#)

finding

[1.2.2. How Linux is Different](#)

[1.2.2.6. Linux distributions](#)

help : [3.2. Getting Help](#)

history of : [1.2.2.4. The Linux kernel](#)

installing : (see [installing Linux](#))

logging in : [4.1.2. Logging In](#)

for the first time : [3.1.4.5. Exiting dselect](#)

modifying : [1.2.2. How Linux is Different](#)

networking and : [10.1. Introduction](#)

obtaining : [1.2.2.6. Linux distributions](#)

origins of : [1.2.2.1. The origins of Linux](#)

vs. other operating systems : [1.2.2. How Linux is Different](#)

performance : [1.2.3. Linux Features and Performance](#)

popularity : [1.1. Linux at Home and at Work](#)

pronunciation of : [1.2. What is Linux?](#)

reasons for choosing/not choosing

[1. Why Run Linux?](#)

[13.3.4. Periscope: A Useful Networking Script](#)

checklist for : [1.3. Reasons to Choose or Not Choose Linux](#)

releases : [1.2.2.4. The Linux kernel](#)

resources for further information

[1.2.1. PC Operating Systems](#)

[1.4. Linux Resources on the Internet](#)

[13.3.6. Continuing Onward](#)

running concurrently with Windows applications : [1.3. Reasons to Choose or Not Choose Linux](#)

as Samba server client : [10.4.3. Linux Client](#)

shells : [13.1.1. A Variety of Shells](#)

power of : [13.2.10. The Power of the Linux Shell](#)

users of : [1.1. Linux at Home and at Work](#)

uses at home/work : [1.1. Linux at Home and at Work](#)

using applications and clients with Debian : [C.6.3. Chapter 8: Using Linux Applications and Clients](#)

using via the X Window System : [6. Using the X Window System](#)

versions of : [1.2.2.4. The Linux kernel](#)

Linux distributions

[1.2.2. How Linux is Different](#)

[1.2.2.6. Linux distributions](#)

[3.1.1. The Installation Program User Interface](#)

Linux Documentation Project : [1.4.1. Web Pages](#)

Linux FAQ : [3.2.2. FAQs](#)

Linux Journal : [13.3.6. Continuing Onward](#)

Linux Journal web site : [1.4.1. Web Pages](#)

Linux Magazine : [13.3.6. Continuing Onward](#)

Linux Mall : [1.3. Reasons to Choose or Not Choose Linux](#)

Linux partition

initializing : [3.1.2.11. Initializing a Linux partition](#)

name of : [3.1.2.9. Partitioning a hard drive](#)

Linux shells : [1.2.3. Linux Features and Performance](#)

Linux Webring web site : [1.4.1. Web Pages](#)

Linux-based LAN (Local Area Network), setting up : (see [LAN](#))

lists

of devices : [4.3.1. Devices](#)

of directories : [4.3.4.3. Displaying directory contents](#)

of files : [4.3.5.1. Displaying the contents of a file](#)

of shell variables : [13.2.7. Shell Variables](#)

of system users : [4.1.3. Issuing Commands](#)

of window managers : [6.3. Window Managers](#)

(lo) loopback device : [13.3.4. Periscope: A Useful Networking Script](#)

Local Area Network (LAN), Linux-based : (see [LAN](#))

local time : [3.1.2.16. Configuring the base system](#)

setting : [7.5.1. Setting the Current System Date and Time](#)

locking (web server) : [12.2.1.3. The httpd.conf File](#)

log files, web server : [12.2.1.3. The httpd.conf File](#)

logging in : [4.1.2. Logging In](#)

login prompt : [4.1.1. Booting the System](#)

logging out : [4.1.6. Logging Out](#)

GNOME desktop : [6.5.2. Logging Out](#)

logical block addressing (LBA) : [2.3.1. How Hard Disks are Organized](#)
logical partitions : [3.1.2.9. Partitioning a hard drive](#)
logical structure (hard disks) : [2.3.1. How Hard Disks are Organized](#)
login group : [4.3.5.9. Working with file permissions](#)
Logitech MouseMan : [5.3. Configuring X](#)
logs, viewing : [7.4. Viewing System Messages and Logs](#)
loopback device (lo) : [13.3.4. Periscope: A Useful Networking Script](#)
loops : [13.3.3.7. The break and continue commands](#)
Lotus 1-2-3 : [8.1.1.1. Running Applixware](#)
Lotus eSuite : [8.1.4. Other Desktop Applications](#)
Lynx : [11.1. Connecting to the Internet](#)
lynx, installing : [C.6.5.1. Installing the lynx package](#)

Index: M

- mail package (exim) : [3.1.4.4. Installing the selected packages](#)
 - configuring after installation : [3.1.4.4. Installing the selected packages](#)
 - log files for : [7.4. Viewing System Messages and Logs](#)
- mail server
 - configuring : [12.3. Using a Mail Server](#)
 - sending/receiving email : [12.3. Using a Mail Server](#)
- mailing lists : [3.2.4. The Debian Mailing Lists](#)
- main menu (GNOME desktop)
 - [6.5.3.6. Main menu](#)
 - [6.7.2. The GNOME Main Menu](#)
- management tools (packages) : [C.2. The Package Management Tools](#)
- manual pages : [4.2.2.1. Using man](#)
- Massachusetts Institute of Technology : (see [MIT](#))
- master user : [3.1.3.1. Establishing a root password](#)
- media, removable : [4.4.1. Mounting and Unmounting Drives](#)
- members
 - adding to groups : [7.1.4.3. Adding a member to a group](#)
 - lists of : [7.1.4.4. Removing a member from a group](#)
 - removing from groups : [7.1.4.4. Removing a member from a group](#)
- memory : [2.1.2. Motherboard](#)
 - configuration information for : [2.2.1. Information You Need](#)
 - configuring with xf86config : [5.3. Configuring X](#)
- menu editor (GNOME desktop) : [6.7.2. The GNOME Main Menu](#)
- Merced 64-bit CPU : [1.2.2. How Linux is Different](#)
- Mesa : [8.2.2. WINE](#)
 - installing : [C.6.3.4. Installing the Mesa and WINE packages](#)
- messages, viewing : [7.4. Viewing System Messages and Logs](#)
- Meta (Alt) key : [C.6.6.4. Installing the minicom package](#)
- meta-information : [C.1. Packages](#)
- metacharacters
 - case condition : [13.3.3.3. The case command](#)
 - filename : [13.2.3. Filename Globbing](#)

for command : [13.3.3.6. The for command](#)
mgetty package : [12.5. Configuring a Dial-In Shell Server](#)
Microsoft Excel : [8.1.1.1. Running Applixware](#)
Microsoft Office 97 : [8.1.2.2. Running StarOffice](#)
Microsoft Office for Linux : [8.1. Linux Desktop Applications](#)
Microsoft Windows : (see entries at Windows)
 games : [9. Playing Linux Games](#)
Microsoft Word : [1.1. Linux at Home and at Work](#)
middle mouse button
 [6.2.1. Copying and Pasting Text](#)
 [6.5.3.2. Desktop](#)
MIME types (GNOME control center) : [6.7.3. The GNOME Control Center](#)
mini-editor (BASH shell) : [13.2.1. Typing Shell Commands](#)
minicom program
 [11.7. Using minicom and seyon](#)
 [11.7.2. Using minicom](#)
 color display : [11.7.2.7. Running minicom](#)
 commands for : [11.7.2.7. Running minicom](#)
 installing : [C.6.6.4. Installing the minicom package](#)
 running : [11.7.2.7. Running minicom](#)
Minix : [1.2.2.4. The Linux kernel](#)
MIRROR : [2.3.3.4. Using the fips utility](#)
MIT (Massachusetts Institute of Technology) : [1.2.2.1. The origins of Linux](#)
MIT (Massachusetts Institute of Technology)
 [1.2.2.5. The X Window System](#)
 [5.1. What is X?](#)
modems : [1.1. Linux at Home and at Work](#)
 configuring : [11.2. Configuring Your Modem](#)
 with minicom program : [11.7.2.3. Configuring the serial port](#)
 for dial-in shell server connections : [12.5. Configuring a Dial-In Shell Server](#)
 options for, configuring (minicom program) : [11.7.2.4. Configuring modem and dialing options](#)
 status indicators : [11.7.1. Using seyon](#)
modemtool : [12.5.1. Configuration](#)

modification date (directory/file) : [4.3.4.3. Displaying directory contents](#)
modules

installing

[3.1.2.12. Installing the Operating System Kernel and Modules](#)

[3.1.2.13. Configuring device driver modules](#)

list of : [3.1.2.13. Configuring device driver modules](#)

monitors

caution when configuring : [5.2. Installing X](#)

configuring with xf86config : [5.3. Configuring X](#)

options for, configuring (minicom program) : [11.7.2.5. Configuring screen and keyboard options](#)

sync rate, caution with (xf86config) : [5.3. Configuring X](#)

monochrome display for installation program display : [3.1.2.4. Choosing color versus monochrome](#)

motherboards : [2.1.2. Motherboard](#)

mount point : [7.2.1. Configuring Local Drives](#)

directory : [4.4.1. Mounting and Unmounting Drives](#)

mounting

CD-ROM : [3.1.4.2. Mounting the CD-ROM](#)

drives : [4.4.1. Mounting and Unmounting Drives](#)

floppy disks : [4.4.2. Formatting Floppy Diskettes](#)

by users : [7.2.1. Configuring Local Drives](#)

partitions automatically : [7.2.1. Configuring Local Drives](#)

mouse : [5.1. What is X?](#)

configuration information for

[2.2.1. Information You Need](#)

[2.2.2. Collecting Configuration Information by Using Windows](#)

configuring with xf86config : [5.3. Configuring X](#)

left mouse button : [6.2.2. Using Scrollbars](#)

Logitech MouseMan : [5.3. Configuring X](#)

not supported for installation program : [3.1.1. The Installation Program User Interface](#)

operations with : [6.2. Mouse Operations](#)

properties (GNOME control center : [6.7.3. The GNOME Control Center](#)

right mouse button : [6.2.2. Using Scrollbars](#)

right-clicking

[6.1.5. Pop-Up Menus](#)

[6.5.3.2. Desktop](#)

[6.5.4. Using the File Manager](#)

three-button, emulating : [5.3. Configuring X](#)

moving files : [4.3.5.4. Renaming or moving a file](#)

MS-DOS prompt

[1.2.3. Linux Features and Performance](#)

[2.3.2. Viewing Partition Information](#)

[4.1.3. Issuing Commands](#)

[6.1.4. Terminal Windows](#)

window : [13.1. The Linux Shell](#)

MS-DOS shell : [13.1.1. A Variety of Shells](#)

Multics : [1.2.2.1. The origins of Linux](#)

multimedia adapters, configuration information for : [2.2.2. Collecting Configuration Information by Using Windows](#)

multimedia applications

GNOME : [6.4.2. GNOME](#)

KDE desktop : [6.4.1. KDE](#)

multimedia files : [12.2.1.2. The srm.conf File](#)

GNOME control center : [6.7.3. The GNOME Control Center](#)

Multiplexed Information and Computing Service : [1.2.2.1. The origins of Linux](#)

My Documents directory : [4.3.4.4. Creating a directory.](#)

Index: N

names

- for directories

 - [4.3.4.3. Displaying directory contents](#)

 - [4.3.4.4. Creating a directory](#)

- for files : [4.3.4.3. Displaying directory contents](#)

- format names (web server) : [12.2.1.3. The httpd.conf File](#)

nameservers : [3.1.2.14. Configuring the network](#)

- establishing : [10.3. Samba](#)

Net BIOS : [10.3. Samba](#)

netbios name option (Samba server) : [10.3.2.1. Configuring global variables](#)

netmask : [3.1.2.14. Configuring the network](#)

- specifying for LAN (Local Area Network) : [10.2.5. Hostname Search Path](#)

Netscape Navigator

- installing : [C.6.6.2. Installing Netscape Navigator](#)

- web browser : [11.5. Web Browser](#)

- web server and : [12.2.2. Startup and Use](#)

Netware file/printer sharin : [10.3. Samba](#)

network adapter cards

- configuration information for

 - [2.2.1. Information You Need](#)

 - [2.2.2. Collecting Configuration Information by Using Windows](#)

- module for, installing : [3.1.2.13. Configuring device driver modules](#)

network address : [3.1.2.14. Configuring the network](#)

network applications

- GNOME : [6.4.2. GNOME](#)

- KDE desktop : [6.4.1. KDE](#)

network mode for playing DOOM : [9.2.1. DOOM](#)

network route, deleting : [13.3.4. Periscope: A Useful Networking Script](#)

network services, restarting : [7.3.2. Starting and Stopping Services](#)

networking script for (periscope) : [13.3.4. Periscope: A Useful Networking Script](#)

networks

configuring : [3.1.2.14. Configuring the network](#)

information needed for : [3.1.2.14. Configuring the network](#)

confirmation screen (configuration) : [3.1.2.14. Configuring the network](#)

network address : [3.1.2.14. Configuring the network](#)

permanent/temporary connection : [3.1.2.14. Configuring the network](#)

new files : [4.3.5.4. Renaming or moving a file](#)

newsgroups

FTP server : [12.1.1. Testing the FTP Server](#)

graphical user interface : [13.1.2. Why Learn to Use the Shell?](#)

Linux : [1.4.2. Newsgroups](#)

Samba server : [10.3.6. Troubleshooting Samba](#)

USENET : [3.2.5. USENET Newsgroups](#)

NFS (Network File System), servers : [2.1.3. Drives](#)

nmbd daemon : [10.3.3. Viewing Samba Server Status](#)

non-ASCII characters : [5.3. Configuring X](#)

Norton Disk Doctor utility : [2.3.3.4. Using the fips utility](#)

Norton Speedisk utility : [2.3.3.4. Using the fips utility](#)

Notepad accessory (Microsoft) : [4.5.2. Using the ae Editor](#)

NTFS (NT filesystem) : [1.2.3. Linux Features and Performance](#)

support for reading : [4.3.2. Filesystems](#)

null files : [13.2.6. Input/Output Redirection and Piping](#)

numerical access mode values : [4.4.2. Formatting Floppy Diskettes](#)

Index: O

O'Callahan, Robert : [12.4.2. Using a Secure Server from a Remote System](#)
obtaining Linux : (see [Linux distributions](#))
Office 97 (Microsoft) : [8.1.2.2. Running StarOffice](#)
Office for Linux (Microsoft) : [8.1. Linux Desktop Applications](#)
office suite (KOffice) : [6.4.1. KDE](#)
old/new files : [4.3.5.4. Renaming or moving a file](#)
open files, viewing : [10.3.3. Viewing Samba Server Status](#)
operating systems : [1.2. What is Linux?](#)
 GNU : [1.2.2.2. Free software](#)
 guest : [8.2.1. The VMware Virtual Platform](#)
 host : [8.2.1. The VMware Virtual Platform](#)
 Linux
 [1.2.1. PC Operating Systems](#)
 [1.2.3. Linux Features and Performance](#)
 sales of for desktop computers : [1.2.1. PC Operating Systems](#)
options : [4.2.1. Command Structure](#)
 description of : [4.2.2. Getting Help](#)
 help for : [4.2.2. Getting Help](#)
OS/2 file/printer sharing : [10.3. Samba](#)
other access permission : [4.3.5.9. Working with file permissions](#)
output data : [13.2.6. Input/Output Redirection and Piping](#)
output redirectors : [13.2.6. Input/Output Redirection and Piping](#)
output stream, standard : [13.2.6. Input/Output Redirection and Piping](#)
owner : [4.3.5.9. Working with file permissions](#)
 access permission : [4.3.5.9. Working with file permissions](#)

Index: P

packages

access methods : [C.4.1. Choosing the Access Method](#)

accessing : [3.1.4.1. Accessing the packages](#)

available, updating list of : [3.1.4.3. Updating the list of available packages](#)

choosing : [3.1.3.6. Choosing Packages](#)

for installation/removal : [C.4.3. Choosing Packages for Installation or Removal](#)

configuring : [C.4.6. Configuring Packages](#)

contents of : [C.6. Package Contents and Installation Commands](#)

database for, querying : [C.3.3. Querying the Package Database](#)

fetchmail package : [12.3. Using a Mail Server](#)

installation commands : [C.6. Package Contents and Installation Commands](#)

installing : [C. The Debian Package Management Utilities](#)

apt-get management tool : [C.5.2.2. Installing a Package](#)

dpkg management tool : [C.3.1. Installing a Package](#)

dselect management tool : [C.4.5. Installing Packages](#)

listing

by name : [C.3.3.2. List packages by name](#)

files installed from : [C.3.3.4. List files installed from a package](#)

for mail server : [12.3. Using a Mail Server](#)

management schemes : [C.1. Packages](#)

management tools for : [C.2. The Package Management Tools](#)

mgetty package : [12.5. Configuring a Dial-In Shell Server](#)

naming conventions : [C.1. Packages](#)

printing description of : [C.3.3.1. Printing the description of a package](#)

recommended/recommending : [C.4.3. Choosing Packages for Installation or Removal](#)

removing

dpkg management tool : [C.3.2. Removing a Package](#)

dselect management tool : [C.4.7. Removing Packages](#)

reporting status of : [C.3.3.3. Report status of a package](#)

selected, installing : [3.1.4.4. Installing the selected packages](#)
suggested/suggesting : [C.4.3. Choosing Packages for Installation or Removal](#)

updating information on

apt-get management tool : [C.5.2.1. Updating Information on Available Packages](#)

dselect management tool : [C.4.2. Updating Information on Available Packages](#)

upgrading (apt-get management tool) : [C.5.2.3. Upgrading Installed Packages](#)

XFree86 : [5.1. What is X?](#)

pager (GNOME desktop)

[6.2.3. Virtual Desktop](#)

[6.5.3.7. Pager](#)

pages, manual : [4.2.2.1. Using man](#)

pair of slashes (//) : [C.5.1. Configuring the sources.list File](#)

panel (GNOME desktop)

[6.5.3.4. Panel](#)

[6.7.1. The GNOME Panel](#)

PAP (Password Authentication Protocol) : [11.3. Using wvdial](#)

parameters : [4.3.5.7. Working with compressed files](#)

partition table : [3.1.2.9. Partitioning a hard drive](#)

PartitionMagic utility : [2.3.3.3. Shrinking a partition](#)

partitions : [2.3.1. How Hard Disks are Organized](#)

creating : [3.1.2.9. Partitioning a hard drive](#)

deleting : [2.3.3.2. Identifying an unused partition](#)

extended : [3.1.2.9. Partitioning a hard drive](#)

identifying unused : [2.3.3.2. Identifying an unused partition](#)

Linux

(see also [Linux partitions](#))

[3.1.2.11. Initializing a Linux partition](#)

initializing : [3.1.2.11. Initializing a Linux partition](#)

logical : [3.1.2.9. Partitioning a hard drive](#)

modifying information : [3.1.2.9. Partitioning a hard drive](#)

mounting additional automatically : [7.2.1. Configuring Local Drives](#)

new, name of : [3.1.2.9. Partitioning a hard drive](#)

primary : [3.1.2.9. Partitioning a hard drive](#)

shrinking : [2.3.3.3. Shrinking a partition](#)

splitting : [2.3.3.3. Shrinking a partition](#)

swap

(see also [swap partitions](#))

[3.1.2.9. Partitioning a hard drive](#)

[3.1.2.10. Initializing and activating a swap partition](#)

verifying newly created : [2.3.3.4. Using the fips utility](#)

viewing information : [2.3.2. Viewing Partition Information](#)

passwords

administering with Samba : [10.3. Samba](#)

authentication : [11.3. Using wvdial](#)

changing : [7.1.3. Changing a User Account Password](#)

encrypted

[C.6.5.2. Installing the samba package](#)

[3.1.3.3. Configuring shadow passwords](#)

insecure : [3.1.3.1. Establishing a root password](#)

managing (Samba server) : [10.3.5. Managing Users and Passwords](#)

restrictions on : [7.1.3. Changing a User Account Password](#)

root : [3.1.3.1. Establishing a root password](#)

security and : [7.1.1. Creating a User Account](#)

shadow

[3.1.3.3. Configuring shadow passwords](#)

[7.1.3. Changing a User Account Password](#)

user accounts : [3.1.3.2. Establishing a normal user account](#)

writing scripts for : [11.7.2.7. Running minicom](#)

pasting text : [6.2.1. Copying and Pasting Text](#)

path

(see also [absolute path](#))

(see also [relative path](#))

[11.7.2.1. Configuring filenames and paths](#)

configuring (minicom program) : [11.7.2.1. Configuring filenames and paths](#)

PCMCIA support, removing : [3.1.3.4. Removing PCMCIA support](#)

PCs : (see [desktop computers](#))

Pentium processors : [2.1.1. Central Processing Unit \(CPU\)](#)

performance of Linux : [1.2.3. Linux Features and Performance](#)

periscope shell script : [13.3.4. Periscope: A Useful Networking Script](#)

scheduling execution : [13.3.4. Periscope: A Useful Networking Script](#)
script file for : [13.3.4. Periscope: A Useful Networking Script](#)
starting : [13.3.4. Periscope: A Useful Networking Script](#)
termination request : [13.3.4. Periscope: A Useful Networking Script](#)
using : [13.3.5. Using periscope](#)
permissions : [4.3.5.9. Working with file permissions](#)
 default (web server) : [12.2.1.1. The access.conf file](#)
 documents/services : [12.2.1.1. The access.conf file](#)
PGP-encrypted email : [8.1.2.2. Running StarOffice](#)
phone numbers : [11.7.2.7. Running minicom](#)
phonelist (seyon program) : [11.7.1. Using seyon](#)
physical structure (hard disks) : [2.3.1. How Hard Disks are Organized](#)
Pico text editor : [4.5.2. Using the ae Editor](#)
 commands for : [4.5.2. Using the ae Editor](#)
pipe redirector (|) : [4.3.4.3. Displaying directory contents](#)
pointing devices
 (see also [mouse](#))
 [5.1. What is X?](#)
 configuring with xf86config : [5.3. Configuring X](#)
 not supported for installation program : [3.1.1. The Installation Program User Interface](#)
pop-up menus : [6.1.5. Pop-Up Menus](#)
POSIX standard : [13.1.1. A Variety of Shells](#)
pound sign (#) : [3.1.4.5. Exiting dselect](#)
power-on self-test : [3.1.2.17. Making a boot floppy.](#)
PowerQuest : [2.3.3.3. Shrinking a partition](#)
PPP (point-to-point protocol) : [11.1. Connecting to the Internet](#)
 log entries about : [7.4. Viewing System Messages and Logs](#)
 login procedures : [11.3. Using wvdial](#)
PPP connection
 [3.1.3.5. Connecting via PPP](#)
 [11. Getting Connected to the Internet](#)
 for periscope shell script : [13.3.4. Periscope: A Useful Networking Script](#)
 manual : [11.8. Making a PPP Connection Manually](#)
 PPP client : [11.4. PPP Client](#)
 PPP server : [11.4. PPP Client](#)

pppd program

[11.1. Connecting to the Internet](#)

[13.3.4. Periscope: A Useful Networking Script](#)

preparation for Linux installation : [2. Preparing to Install Linux](#)

presentation graphics applications : [8.1.4. Other Desktop Applications](#)

StarImpress : [8.1.2.2. Running StarOffice](#)

Presents (Applixware) : [8.1.1.1. Running Applixware](#)

primary network interface : [3.1.2.14. Configuring the network](#)

primary partitions : [3.1.2.9. Partitioning a hard drive](#)

print job number : [4.3.5.6. Printing a file](#)

print shares : [10.4.1. Microsoft Windows Client](#)

active, viewing : [10.3.3. Viewing Samba Server Status](#)

options for Samba server : [10.3.2.1. Configuring global variables](#)

parameters for Samba server, configuring : [10.3.2.3. Configuring printer share parameters](#)

printer sharing : [10.3. Samba](#)

Printers tool bar button (Samba server) : [10.3.2.3. Configuring printer share parameters](#)

printers, shared : [10.4.1. Microsoft Windows Client](#)

printing

cancel : [4.3.5.6. Printing a file](#)

files

[4.3.5.6. Printing a file](#)

[10.4.3. Linux Client](#)

probing systems

caution with : [5.3. Configuring X](#)

with xf86config : [5.3. Configuring X](#)

process id (web server) : [12.2.1.3. The httpd.conf File](#)

.profile file : [13.2.5. Shell Scripts](#)

profiles : [3.1.3.6. Choosing Packages](#)

programmable clock chips : [5.3. Configuring X](#)

programming languages : [1.2.2.1. The origins of Linux](#)

shells and : [13.2.7. Shell Variables](#)

programs

finding via search path : [13.2.8. The Search Path](#)

running

[4.3.5.10. Running programs](#)

[13.2.2. Commands and Arguments](#)

search path and : [13.2.8. The Search Path](#)

Project Athena : [1.2.2.5. The X Window System](#)

prompting for confirmation (commands) : [13.2.4. Shell Aliases](#)

prompts

FTP : [12.1.1. Testing the FTP Server](#)

FTP subsystem : [12.1.1. Testing the FTP Server](#)

MS-DOS

[1.2.3. Linux Features and Performance](#)

[2.3.2. Viewing Partition Information](#)

[4.1.3. Issuing Commands](#)

[6.1.4. Terminal Windows](#)

secondary prompts : [13.3.3.2. The if command](#)

shell : [3.1.4.5. Exiting dselect](#)

pronunciation of Linux : [1.2. What is Linux?](#)

protecting data : [2.3.3.4. Using the fips utility](#)

protocol, SMB (Server Message Block) : [10.3. Samba](#)

proxy server (web server) : [12.2.1.3. The httpd.conf File](#)

pseudodevices : [4.3.1. Devices](#)

Index: Q

Quadratron Cliq : [8.1.4. Other Desktop Applications](#)

Quake II : [9.2.2. Quake II](#)

installing : [C.6.4.2. Installing the quake2 package](#)

queueing files : [4.3.5.6. Printing a file](#)

quote characters : [13.2.9. Quoted Strings](#)

Index: R

RAID : [1.2.3. Linux Features and Performance](#)

RAM memory : [2.1.2. Motherboard](#)

configuration information for : [2.2.1. Information You Need](#)

RAMDAC settings : [5.3. Configuring X](#)

read/write heads : [2.3.1. How Hard Disks are Organized](#)

readme entries : [12.2.1.2. The srm.conf File](#)

rebooting your system

[2.3.3.4. Using the fips utility.](#)

[7.3.1. Starting and Stopping the System](#)

under the X Window System : [6.1.3. Terminating X](#)

recycle bin : [6.5.4. Using the File Manager](#)

Red Hat Linux : [1.2.2.6. Linux distributions](#)

redirectors : [13.2.6. Input/Output Redirection and Piping](#)

ignored by programs : [13.2.6. Input/Output Redirection and Piping](#)

pipe redirector (|) : [13.2.6. Input/Output Redirection and Piping](#)

relational database : [8.1.4. Other Desktop Applications](#)

relative parameters : [4.3.5.7. Working with compressed files](#)

relative path

[4.3.3.3. Absolute and relative pathnames](#)

[4.3.4.4. Creating a directory.](#)

release notes for installation program : [3.1.2.5. Release notes](#)

releases of Debian. mailing list information : [3.2.4. The Debian Mailing Lists](#)

remote mail protocol : [12.3. Using a Mail Server](#)

removable media : [4.4.1. Mounting and Unmounting Drives](#)

Remove item (dselect program) : [3.1.4. Installing the Applications](#)

removing directories : [4.3.4.5. Removing a directory.](#)

removing files : [4.3.5.2. Removing a file](#)

removing packages

dpkg management tool : [C.3.2. Removing a Package](#)

dselect management tool : [C.4.7. Removing Packages](#)

pre-selection : [C.4.3. Choosing Packages for Installation or Removal](#)

renaming

autoexec.bat file : [2.3.3.4. Using the fips utility](#)
config.sys file : [2.3.3.4. Using the fips utility](#)
directories : [4.3.5.4. Renaming or moving a file](#)
files

[4.3.5.4. Renaming or moving a file](#)

[6.5.4. Using the File Manager](#)

using links : [4.3.5.8. Working with links](#)

Replay Associates : [12.4. Configuring a Secure Shell Server](#)

resources for further information : [8.1.1. Applix Appixware](#)

advanced encryption : [12.4. Configuring a Secure Shell Server](#)

Apache (web server) : [12.2.1. Configuration](#)

desktop applications : [8.1.4. Other Desktop Applications](#)

DOOM (game) : [9.2.1. DOOM](#)

FTP server : [12.1.1. Testing the FTP Server](#)

GNOME desktop : [6.4.2. GNOME](#)

GNOME desktop applets and applications : [6.6. Using GNOME Applets and Applications](#)

id Software : [9.2.1. DOOM](#)

KDE desktop : [6.4.1. KDE](#)

Linux

[1.2.1. PC Operating Systems](#)

[1.4. Linux Resources on the Internet](#)

[13.3.6. Continuing Onward](#)

printing

with smbprint script : [10.4.3. Linux Client](#)

Samba : [10.3. Samba](#)

secure shell server : [12.4. Configuring a Secure Shell Server](#)

Starcraft (game) : [9.2.3.1. Installing Starcraft](#)

StarOffice desktop suite : [8.1.2. StarOffice](#)

VMware : [8.2.1. The VMware Virtual Platform](#)

window managers : [6.3. Window Managers](#)

WINE project : [8.2.2. WINE](#)

WordPerfect for Linux (Corel) : [8.1.3. Corel WordPerfect for Linux](#)

X11 interface : [10.4.3. Linux Client](#)

restarting network services : [7.3.2. Starting and Stopping Services](#)

restarting your system : [7.3.1. Starting and Stopping the System](#)

restoring backups : [10.4.4. Using the Linux Samba Client for File Backup and Recovery](#)

restorrb program : [2.3.3.4. Using the fips utility](#)

Richie, Dennis : [1.2.2.1. The origins of Linux](#)

right mouse button

[6.2.1. Copying and Pasting Text](#)

[6.2.2. Using Scrollbars](#)

right-clicking (mouse)

[6.1.5. Pop-Up Menus](#)

[6.5.3.2. Desktop](#)

[6.5.4. Using the File Manager](#)

RNG (modem indicator) : [11.7.1. Using seyon](#)

root (user)

[C.2. The Package Management Tools](#)

[3.1.3.1. Establishing a root password](#)

[4.4.1. Mounting and Unmounting Drives](#)

[5.2. Installing X](#)

account for : [3.1.3.2. Establishing a normal user account](#)

root directory : [4.3.3.2. The directory tree](#)

root password : [3.1.3.1. Establishing a root password](#)

routers : (see [gateways](#))

RTF file format : [8.1.1.1. Running Applixware](#)

RTS (modem indicator) : [11.7.1. Using seyon](#)

running programs : [4.3.5.10. Running programs](#)

search path and : [13.2.8. The Search Path](#)

Index: S

safety/security, specifying options for : [10.3.2.1. Configuring global variables](#)

Samba : [10.3. Samba](#)

Samba server

backups : [10.3.6. Troubleshooting Samba](#)

client configuration : [10.4. Samba Client Configuration and Use](#)

configuring : [10.3.2. Configuring Samba](#)

global variables, configuring : [10.3.2.1. Configuring global variables](#)
installing

[C.6.5.2. Installing the samba package](#)

[10.3.1. Samba Server Installation](#)

log entries about : [7.4. Viewing System Messages and Logs](#)

netbios name option : [10.3.2.1. Configuring global variables](#)

print share options for : [10.3.2.1. Configuring global variables](#)

printer share parameters, configuring : [10.3.2.3. Configuring printer share parameters](#)

resources for further information : [10.3. Samba](#)

troubleshooting : [10.3.6. Troubleshooting Samba](#)

verifying operation of : [10.3.6. Troubleshooting Samba](#)

version of, viewing : [10.3.3. Viewing Samba Server Status](#)

viewing configuration : [10.3.5. Managing Users and Passwords](#)

viewing status of : [10.3.3. Viewing Samba Server Status](#)

workgroup option : [10.3.2.1. Configuring global variables](#)

saving changes (minicom program) : [11.7.2.6. Saving your changes](#)

/sbin file : [13.2.2. Commands and Arguments](#)

ScanDisk utility : [2.3.3.4. Using the fips utility](#)

schedulers : [8.1.4. Other Desktop Applications](#)

StarSchedule : [8.1.2.2. Running StarOffice](#)

scientists : [1.1. Linux at Home and at Work](#)

screen options, configuring (minicom program) : [11.7.2.5. Configuring screen and keyboard options](#)

screen output

viewing directories : [4.3.4.3. Displaying directory contents](#)

viewing files : [4.3.5.1. Displaying the contents of a file](#)
viewing tar files : [4.3.5.7. Working with compressed files](#)
screen savers : [4.1.5. Using Virtual Consoles](#)
GNOME control center : [6.7.3. The GNOME Control Center](#)
screens : (see [monitors](#))
script file for periscope shell script : [13.3.4. Periscope: A Useful Networking Script](#)
scripts
 [11.7.2.1. Configuring filenames and paths](#)
 [11.7.2.7. Running minicom](#)
modifying standard : [13.2.5. Shell Scripts](#)
shell scripts
 [13.2.5. Shell Scripts](#)
 [13.2.6. Input/Output Redirection and Piping](#)
 advanced : [13.3. Understanding Shell Scripts](#)
 periscope : [13.3.4. Periscope: A Useful Networking Script](#)
 standard included with Linux : [13.2.5. Shell Scripts](#)
scroll bars : [6.2.2. Using Scrollbars](#)
SCSI adapters
 configuration information for : [2.2.1. Information You Need](#)
search engine for Linux information : [1.4.1. Web Pages](#)
search path
 [13.2.8. The Search Path](#)
 [13.3.4. Periscope: A Useful Networking Script](#)
searching by keywords : [4.2.2.2. Using apropos](#)
secondary prompt : [13.3.3.2. The if command](#)
sectors : [2.3.1. How Hard Disks are Organized](#)
secure shell client
 [C.6.7.2. Installing the ssh client and server package](#)
 [12.5. Configuring a Dial-In Shell Server](#)
secure shell server : [C.6.7.2. Installing the ssh client and server package](#)
 configuring : [12.4. Configuring a Secure Shell Server](#)
 secure shell client on : [12.4.2. Using a Secure Server from a Remote System](#)
security
 mailing list information : [3.2.4. The Debian Mailing Lists](#)
 passwords and : [7.1.1. Creating a User Account](#)

specifying options for : [10.3.2.1. Configuring global variables](#)
Select item (dselect program) : [3.1.4. Installing the Applications](#)
self-test : [3.1.2.17. Making a boot floppy](#)
serial port, configuring (minicom program) : [11.7.2.3. Configuring the serial port](#)
server administrator, email address of : [12.2.1.3. The httpd.conf File](#)
server daemons : [10.3.3. Viewing Samba Server Status](#)
server logs, web server : [12.2.1.3. The httpd.conf File](#)
Server Message Block (SMB)

[10.3. Samba](#)

[10.4.1. Microsoft Windows Client](#)

server processes (web server) : [12.2.1.3. The httpd.conf File](#)
servers

Linux as

[1.2.1. PC Operating Systems](#)

[1.2.3. Linux Features and Performance](#)

Samba : (see [Samba server](#))

SMB servers : [10.4.3. Linux Client](#)

webserver for Linux : (see [web server](#))

X server, configuring : [5.3. Configuring X](#)

services

list of : [7.3.2. Starting and Stopping Services](#)

permissions (web server) : [12.2.1.1. The access.conf file](#)

starting/stopping : [7.3.2. Starting and Stopping Services](#)

session awareness : [6.4.2. GNOME](#)

seyon program : [11.7.1. Using seyon](#)

installing : [C.6.6.3. Installing the seyon package](#)

running : [11.7.1. Using seyon](#)

sh shell : [7.1.6. Configuring Access to Shells](#)

shadow passwords

[3.1.3.3. Configuring shadow passwords](#)

[7.1.3. Changing a User Account Password](#)

shared printers, using : [10.4.1. Microsoft Windows Client](#)

shares

(see

(see [browseable shares](#))

(see [print shares](#))

(see entries at file shares)

Shares button (Samba server) : [10.3.2.2. Configuring file share parameters](#)

shareware programs : [2.3.3.4. Using the fips utility](#)

shell accounts : [11.1. Connecting to the Internet](#)
accessing : [11.7. Using minicom and seyon](#)

shell aliases
[13. Conquering the BASH Shell](#)
[13.2.4. Shell Aliases](#)

shell commands
[6.6.1. GNOME Terminal](#)
[13.2.1. Typing Shell Commands](#)
aliases for : [13.2.4. Shell Aliases](#)

shell prompt : [3.1.4.5. Exiting dselect](#)

shell scripts
[13. Conquering the BASH Shell](#)
[13.1.2. Why Learn to Use the Shell?](#)
[13.2.5. Shell Scripts](#)
[13.2.6. Input/Output Redirection and Piping](#)
advanced : [13.3. Understanding Shell Scripts](#)
periscope : [13.3.4. Periscope: A Useful Networking Script](#)
special charactes for : [13.2.1. Typing Shell Commands](#)

shell servers
(see [dial-in shell server](#))
(see entries at secure shell server)

shell variables : [13. Conquering the BASH Shell](#)
list of : [13.2.7. Shell Variables](#)
value of : [13.2.7. Shell Variables](#)

shells : [7.1.6. Configuring Access to Shells](#)
assigned automatically : [7.1.6. Configuring Access to Shells](#)
available : [7.1.6. Configuring Access to Shells](#)

bash shell
(see also [BASH shell](#))
[4.1.2. Logging In](#)
changing : [7.1.6. Configuring Access to Shells](#)
characters, special : [13.2.1. Typing Shell Commands](#)
exiting from new shell : [13.2.5. Shell Scripts](#)
MS-DOS shell : [13.1.1. A Variety of Shells](#)

programming languages and : [13.2.7. Shell Variables](#)
shell aliases : [13. Conquering the BASH Shell](#)
shell scripts : [13. Conquering the BASH Shell](#)
shell variables : [13. Conquering the BASH Shell](#)
system administrators and : [13.1.2. Why Learn to Use the Shell?](#)
Unix shell : [13.1.1. A Variety of Shells](#)
variables for

[13.2.7. Shell Variables](#)

[13.3.1. Processing Arguments](#)

vs. Microsoft Windows operating system : [13.1.2. Why Learn to Use the Shell?](#)

Shortcut to boot.bat desktop icon : [3.1.2.1. Booting from MS-DOS or Windows 9x](#)

shortcuts : [4.3.5.8. Working with links](#)

shrinking partitions : [2.3.3.3. Shrinking a partition](#)

shutting down the X Window System : [6.1.3. Terminating X](#)

shutting down Windows 95/98 : [4.1.1. Booting the System](#)

shutting down your system

[4.1.7. Shutting Down the System](#)

[7.3.1. Starting and Stopping the System](#)

single dot (.) : [4.3.3.3. Absolute and relative pathnames](#)

single quote (')

[4.3.4.4. Creating a directory](#)

[13.2.9. Quoted Strings](#)

size (directory/file) : [4.3.4.3. Displaying directory contents](#)

Slackware Linux : [1.2.2.6. Linux distributions](#)

slash (/)

[C.4.3. Choosing Packages for Installation or Removal](#)

[C.5.1. Configuring the sources.list File](#)

pair of slashes (//) : [C.5.1. Configuring the sources.list File](#)

Slashdot web site : [1.4.1. Web Pages](#)

SMB (Server Message Block)

[10.3. Samba](#)

[10.4.1. Microsoft Windows Client](#)

SMB clients

[10.4.2. Other Clients](#)

[10.4.4. Using the Linux Samba Client for File Backup and Recovery](#)

SMB servers

(see also [Samba server](#))

[10.3.2.1. Configuring global variables](#)

[10.4.3. Linux Client](#)

SMB shared volume : [2.1.3. Drives](#)

smbd daemon

[10.3.1. Samba Server Installation](#)

[10.3.3. Viewing Samba Server Status](#)

smbprint script : [10.4.3. Linux Client](#)

Snes9X : [9.1. A Survey of Linux Games](#)

soft links : [4.3.5.8. Working with links](#)

software development : [1.2.3. Linux Features and Performance](#)

software distribution

[1.2.2. How Linux is Different](#)

[1.2.2.2. Free software](#)

[1.2.2.3. Copyleft](#)

Linux : (see [Linux distributions](#))

terms of : [1.2.2.3. Copyleft](#)

Software in the Public Interest, Inc. : [1.2.2.6. Linux distributions](#)

Solaris 2.6, compared : [1.2.3. Linux Features and Performance](#)

sound cards : [2.2.2. Collecting Configuration Information by Using Windows](#)

sounds (GNOME control center) : [6.7.3. The GNOME Control Center](#)

source form/code : [1.2.2. How Linux is Different](#)

source list, changing : [3.1.4.1. Accessing the packages](#)

sources : [C.2. The Package Management Tools](#)

sources.list file : [C.5. Using apt-get](#)

space separating arguments/options : [4.2.1. Command Structure](#)

splitting partitions : [2.3.3.3. Shrinking a partition](#)

spreadsheets : [8.1.4. Other Desktop Applications](#)

Spreadsheets (Applixware) : [8.1.1.1. Running Applixware](#)

srm.conf file : [12.2.1.2. The srm.conf File](#)

ssh client/server, installing : [C.6.7.2. Installing the ssh client and server package](#)

SSH Protocols : [12.4. Configuring a Secure Shell Server](#)

Stallman, Richard : [1.2.2.2. Free software](#)

Starcraft : [9.2.3. Starcraft](#)

StarDraw (StarOffice) : [8.1.2.2. Running StarOffice](#)
StarImpress (StarOffice) : [8.1.2.2. Running StarOffice](#)
StarOffice
installing
[C.6.3.1. Installing Star Office](#)
[8.1.2.1. Installing StarOffice](#)
running : [8.1.2.2. Running StarOffice](#)
StarOffice (Star Division) : [8.1.2. StarOffice](#)
StarSchedule (StarOffice) : [8.1.2.2. Running StarOffice](#)
Start menu (Windows 95/98) : [6.5.3.7. Pager](#)
starting/stopping services : [7.3.2. Starting and Stopping Services](#)
starting/stopping the X Window System : [5.4. Starting and Stopping X](#)
starting/stopping your system : [7.3. Starting and Stopping the System and Services](#)
StarWriter (StarOffice) : [8.1.2.2. Running StarOffice](#)
Status button (swat tool) : [10.3.3. Viewing Samba Server Status](#)
stderr : [13.2.6. Input/Output Redirection and Piping](#)
stdin : [13.2.6. Input/Output Redirection and Piping](#)
stdout : [13.2.6. Input/Output Redirection and Piping](#)
streams : [13.2.6. Input/Output Redirection and Piping](#)
subdirectories : [4.3.4.5. Removing a directory](#)
subnetworks (subnets) : [10.1. Introduction](#)
Sun Microsystems : [8.1.2. StarOffice](#)
SunOS : [1.2.2.1. The origins of Linux](#)
SuSE. Linux : [1.2.2.6. Linux distributions](#)
swap filesystem : [7.2.1. Configuring Local Drives](#)
swap partitions : [3.1.2.9. Partitioning a hard drive](#)
activating/initializing : [3.1.2.10. Initializing and activating a swap partition](#)
configuring : [7.2.2. Configuring Swap Partitions](#)
swat tool
[C.6.5.2. Installing the samba package](#)
[10.3.2. Configuring Samba](#)
accessing : [10.3.2. Configuring Samba](#)
symbolic link, configuring with xf86config : [5.3. Configuring X](#)
symbolic links
[4.3.5.8. Working with links](#)

[5.2. Installing X](#)

system administration commands : [13.2.2. Commands and Arguments](#)

system BIOS : [2.2.1. Information You Need](#)

system buses : [2.1.2. Motherboard](#)

system clocks, Unix : [3.1.2.16. Configuring the base system](#)

system commands : [4.5.1. Viewing System Information](#)

System Properties dialog box

[2.2.2. Collecting Configuration Information by Using Windows](#)

[2.3.3.4. Using the fips utility](#)

system status

displaying : [4.1.3. Issuing Commands](#)

viewing information on : [4.5.1. Viewing System Information](#)

system use cycle : [4.1. The System Use Cycle](#)

System V : [1.2.2.1. The origins of Linux](#)

shell : [7.1.6. Configuring Access to Shells](#)

system, rebooting : [3.1.2.17. Making a boot floppy](#)

Index: T

Tab keystroke : [13.2.1. Typing Shell Commands](#)
tab separating arguments/options : [4.2.1. Command Structure](#)
Tannenbaum, Andrew : [1.2.2.4. The Linux kernel](#)
tar files : [4.3.5.7. Working with compressed files](#)
 creating : [4.3.5.7. Working with compressed files](#)
 StarOffice : [8.1.2.1. Installing StarOffice](#)
tasks : [3.1.3.6. Choosing Packages](#)
TCP/IP
 [1.3. Reasons to Choose or Not Choose Linux](#)
 [12.4. Configuring a Secure Shell Server](#)
tcsh shell : [7.1.6. Configuring Access to Shells](#)
telnet vs. secure shell server : [12.4. Configuring a Secure Shell Server](#)
terminal windows : [6.1.4. Terminal Windows](#)
termination request for periscope shell script : [13.3.4. Periscope: A Useful Networking Script](#)
TeX : [8.1.4. Other Desktop Applications](#)
text editor (Pico) : [4.5.2. Using the ae Editor](#)
text editor, default (GNOME control center) : [6.7.3. The GNOME Control Center](#)
text-mode programs : [5.3. Configuring X](#)
TFP prompt : [12.1.1. Testing the FTP Server](#)
Thompson, Ken : [1.2.2.1. The origins of Linux](#)
thumbnails : [6.2.3. Virtual Desktop](#)
time
 applet for (GNOME desktop) : [6.5.3.5. Date & time applet](#)
 displaying : [4.1.3. Issuing Commands](#)
 setting
 [3.1.2.16. Configuring the base system](#)
 [7.5. Setting the System Time and Time Zone](#)
 storing : [7.5. Setting the System Time and Time Zone](#)
 Universal Time (GMT) : [3.1.2.16. Configuring the base system](#)
time zone
 selecting : [3.1.2.16. Configuring the base system](#)

setting : [7.5.2. Setting the Time Zone](#)
timeouts (web server) : [12.2.1.3. The httpd.conf File](#)
token ring : [3.1.2.14. Configuring the network](#)
Torvalds, Linus
 [1.2. What is Linux?](#)
 [1.2.2.4. The Linux kernel](#)
tracks : [2.3.1. How Hard Disks are Organized](#)
Tridgell, Andrew : [10.3. Samba](#)
troubleshooting
 devices : [3.1.2.13. Configuring device driver modules](#)
 Samba server : [10.3.6. Troubleshooting Samba](#)
 system startup sequence : [7.4. Viewing System Messages and Logs](#)
Try button (GNOME control center) : [6.7.3. The GNOME Control Center](#)
ttssh remote client : [12.4.2. Using a Secure Server from a Remote System](#)
two dots (..) : [4.3.3.3. Absolute and relative pathnames](#)
type (directories/files) : [4.3.4.3. Displaying directory contents](#)

Index: U

uncompressing/unzipping files : [4.3.5.7. Working with compressed files](#)

underscore (_) : [4.3.4.4. Creating a directory](#)

uninstalling packages

dpkg management tool : [C.3.2. Removing a Package](#)

dselect management tool : [C.4.7. Removing Packages](#)

universal resource identifier (URI) : [C.5.1. Configuring the sources.list File](#)

Universal Time (GMT) : [3.1.2.16. Configuring the base system](#)

Universal Time (UTC), setting : [7.5.1. Setting the Current System Date and Time](#)

Unix

[1.2.2.1. The origins of Linux](#)

[1.3. Reasons to Choose or Not Choose Linux](#)

shell variables and : [13.2.7. Shell Variables](#)

shells used by Linux : [13.1.1. A Variety of Shells](#)

standard userid : [12.2.1.3. The httpd.conf File](#)

system clocks : [3.1.2.16. Configuring the base system](#)

unmounting

devices : [7.2.2. Configuring Swap Partitions](#)

drives : [4.4.1. Mounting and Unmounting Drives](#)

floppy disks : [4.4.2. Formatting Floppy Diskettes](#)

until condition : [13.3.3.5. The until command](#)

Up key : [13.2.1. Typing Shell Commands](#)

Update step (dselect program) : [3.1.4. Installing the Applications](#)

upgrading packages : [C. The Debian Package Management Utilities](#)

apt-get management tool : [C.5.2.3. Upgrading Installed Packages](#)

uploads

[11.7.2.1. Configuring filenames and paths](#)

[11.7.2.2. Configuring file transfer protocols](#)

URI (universal resource identifier) : [C.5.1. Configuring the sources.list File](#)

URLs

advanced encryption : [12.4. Configuring a Secure Shell Server](#)

Apache : [12.2.1. Configuration](#)
Applixware : [8.1.1. Applix Applixware](#)
CD help : [6.6.2. GNOME CD Player](#)
Debian GNU/Linux web site : [3.2.1. The Debian Web Site](#)
Debian/GNU/Linux FAQ : [3.2.2. FAQs](#)
Doom
 downloading : [C.6.4.1. Installing Doom](#)
Doom (game) : [9.2.1. DOOM](#)
GNOME desktop : [6.4.2. GNOME](#)
GNOME desktop applets and applications : [6.6. Using GNOME Applets and Applications](#)
hardware requirements for Debian GNU/Linux : [2.1. Minimum Hardware Requirements](#)
id Software : [9.2.1. DOOM](#)
KDE desktop : [6.4.1. KDE](#)
Linux : [1.4.1. Web Pages](#)
Linux distributions, obtaining : [1.2.2.6. Linux distributions](#)
Linux games : [9.1. A Survey of Linux Games](#)
Linux HOWTOs : [3.2.3. HOWTOs](#)
Linux Mall : [1.3. Reasons to Choose or Not Choose Linux](#)
mailing lists : [3.2.4. The Debian Mailing Lists](#)
Netscape Navigator, downloading : [C.6.6.2. Installing Netscape Navigator](#)
package management schemes : [C.1. Packages](#)
pronunciation of Linux : [1.2. What is Linux?](#)
Samba : [10.3. Samba](#)
secure shell client : [12.5. Configuring a Dial-In Shell Server](#)
secure shell server : [12.4. Configuring a Secure Shell Server](#)
Starcraft (game) : [9.2.3.1. Installing Starcraft](#)
StarOffice : [8.1.2. StarOffice](#)
swat tool : [C.6.5.2. Installing the samba package](#)
USENET newsgroups : [3.2.5. USENET Newsgroups](#)
VMware : [8.2.1. The VMware Virtual Platform](#)
Voodoo acceleration : [C.6.4.2. Installing the quake2 package](#)
window managers : [6.3. Window Managers](#)
WINE emulation of Windows applications : [1.3. Reasons to Choose or Not Choose Linux](#)

- WINE project : [8.2.2. WINE](#)
- WordPerfect for Linux (Corel) : [8.1.3. Corel WordPerfect for Linux](#)
- X11 interface : [10.4.3. Linux Client](#)
- USENET newsgroups : [3.2.5. USENET Newsgroups](#)
- user accounts
 - creating with no password : [7.1.1. Creating a User Account](#)
 - deleting : [7.1.5. Deleting a User Account](#)
 - establishing : [3.1.3.2. Establishing a normal user account](#)
- user interface
 - controls : [3.1.1.1. User-interface controls](#)
 - for installation program : [3.1.1. The Installation Program User Interface](#)
- /usr/local/bin file : [13.2.2. Commands and Arguments](#)
- userids : [4.1.2. Logging In](#)
 - FTP server : [12.1.1. Testing the FTP Server](#)
 - Samba server
 - [10.3.1. Samba Server Installation](#)
 - [10.3.5. Managing Users and Passwords](#)
 - standard userid for Unix : [12.2.1.3. The httpd.conf File](#)
 - web server : [12.2.1.3. The httpd.conf File](#)
 - writing scripts for : [11.7.2.7. Running minicom](#)
- usernames
 - changing : [7.1.2. Changing a User's Name](#)
 - creating : [3.1.3.2. Establishing a normal user account](#)
- users
 - accounts for : [3.1.3.2. Establishing a normal user account](#)
 - administering : [7.1. User and Group Administration](#)
 - with Samba : [10.3. Samba](#)
 - authentication
 - [11.3. Using wvdial](#)
 - [12.1. An FTP Server](#)
 - changing their own password : [7.1.3. Changing a User Account Password](#)
 - displaying list of : [4.1.3. Issuing Commands](#)
 - managing (Samba server) : [10.3.5. Managing Users and Passwords](#)
 - permissions for : [12.2.1.1. The access.conf file](#)
 - when mounting floppy disks : [7.2.1. Configuring Local Drives](#)

using smbprint scrip : [10.4.3. Linux Client](#)
/usr/bin filename : [13.2.2. Commands and Arguments](#)
/usr/doc directory : [12.2.1.1. The access.conf file](#)
/usr/doc/minicom/demos : [11.7.2.7. Running minicom](#)
/usr/doc/minicom/doc : [11.7.2.7. Running minicom](#)
/usr/sbin file : [13.2.2. Commands and Arguments](#)
UTC (Universal Time), setting : [7.5.1. Setting the Current System Date and Time](#)

Index: V

- /var/log/messages file : [7.4. Viewing System Messages and Logs](#)
- variables for scripts : [13.3.1. Processing Arguments](#)
- variables for shells : [13.2.7. Shell Variables](#)
- verifying newly created partitions : [2.3.3.4. Using the fips utility](#)
- vertical refresh rate (xf86config) : [5.3. Configuring X](#)
- video adapters : (see [video cards](#))
- video cards : [2.2.2. Collecting Configuration Information by Using Windows](#)
 - configuration information for : [2.2.1. Information You Need](#)
 - configuring with xf86config : [5.3. Configuring X](#)
- video modes
 - switching between : [6.1.1. Switching Video Modes](#)
- View button (swat tool) : [10.3.4. Viewing Samba Server Configuration](#)
- viewing logs/messages : [7.4. Viewing System Messages and Logs](#)
- viewing screen output
 - directories : [4.3.4.3. Displaying directory contents](#)
 - files : [4.3.5.1. Displaying the contents of a file](#)
 - tar files : [4.3.5.7. Working with compressed files](#)
- viewing system status
 - commands for : [4.5.1. Viewing System Information](#)
- virtual consoles : [4.4.1. Mounting and Unmounting Drives](#)
 - CD-ROM, mounting : [3.1.4.2. Mounting the CD-ROM](#)
 - switching to from the X Window System : [6.1.2. Using Virtual Consoles with X](#)
- using
 - [3.1.1.3. Using virtual consoles](#)
 - [4.1.5. Using Virtual Consoles](#)
- virtual desktops
 - [6.2.3. Virtual Desktop](#)
 - [6.5.3.7. Pager](#)
- virtual hosts
 - establishing (web server) : [12.2.1.3. The httpd.conf File](#)
 - web server : [12.2.1.3. The httpd.conf File](#)

virtual memory

 disabling : [2.3.3.4. Using the fips utility](#)

 enabling : [2.3.3.4. Using the fips utility](#)

virtual platforms : [8.2.1. The VMware Virtual Platform](#)

VMware for Linux : [8.2.1. The VMware Virtual Platform](#)

VMware for Windows NT : [8.2.1. The VMware Virtual Platform](#)

VMware, Inc. : [8.2.1. The VMware Virtual Platform](#)

Volosuk, Ivan : [10.4.3. Linux Client](#)

Voodoo acceleration : [C.6.4.2. Installing the quake2 package](#)

Index: W

WAN (Wide-Area Network), Linux-based, setting up

[C.6.7.1. Installing the apache package](#)

[12. Setting Up a Linux-Based WAN](#)

web browsers

[6.5.3.9. Help viewer](#)

[11.5. Web Browser](#)

caching documents and : [12.2.1.3. The httpd.conf File](#)

Netscape Navigator : [11.5. Web Browser](#)

problems with : [12.2.1.2. The srm.conf File](#)

web pages : (see [URLs](#))

web server

binding : [12.2.1.3. The httpd.conf File](#)

caching documents : [12.2.1.3. The httpd.conf File](#)

canonical URL : [12.2.1.3. The httpd.conf File](#)

child processes : [12.2.1.3. The httpd.conf File](#)

configuration files : [12.2.1. Configuration](#)

access.conf file : [12.2.1.1. The access.conf file](#)

httpd.conf file : [12.2.1.3. The httpd.conf File](#)

srm.conf file : [12.2.1.2. The srm.conf File](#)

connections, persistent : [12.2.1.3. The httpd.conf File](#)

error file : [12.2.1.3. The httpd.conf File](#)

format names : [12.2.1.3. The httpd.conf File](#)

groups : [12.2.1.3. The httpd.conf File](#)

hostnames : [12.2.1.3. The httpd.conf File](#)

hosts, establishing : [12.2.1.3. The httpd.conf File](#)

installing/configuring : [12.1.1. Testing the FTP Server](#)

IP address : [12.2.1.3. The httpd.conf File](#)

locking : [12.2.1.3. The httpd.conf File](#)

log files : [12.2.1.3. The httpd.conf File](#)

permissions : [12.2.1.1. The access.conf file](#)

proxy server : [12.2.1.3. The httpd.conf File](#)

server logs : [12.2.1.3. The httpd.conf File](#)

server processes : [12.2.1.3. The httpd.conf File](#)

starting : [12.2.2. Startup and Use](#)
timeouts : [12.2.1.3. The httpd.conf File](#)
URL for : [12.2.2. Startup and Use](#)
userids : [12.2.1.3. The httpd.conf File](#)
using : [12.2.2. Startup and Use](#)
verifying operation : [12.2.2. Startup and Use](#)
virtual hosts : [12.2.1.3. The httpd.conf File](#)
while condition : [13.3.3.4. The while command](#)
Wide-Area Network (WAN) : (see [WAN](#))
wildcards (MS-DOS) : [13.2.3. Filename Globbing](#)
window managers
 [1.2.2.5. The X Window System](#)
 [6. Using the X Window System](#)
 [6.3. Window Managers](#)
 [6.5.1. Launching GNOME and Enlightenment](#)
 list of : [6.3. Window Managers](#)
 selecting (GNOME control center) : [6.7.3. The GNOME Control Center](#)
Windows 2000 Pro (beta)
 running simultaneously with Linux : [8.2. Other Approaches to Desktop Computing](#)
Windows 3.1
 running simultaneously with Linux : [8.2. Other Approaches to Desktop Computing](#)
 as Samba server client : [10.4.1. Microsoft Windows Client](#)
Windows 95/98
 [1.2.1. PC Operating Systems](#)
 [1.2.2. How Linux is Different](#)
 [1.2.3. Linux Features and Performance](#)
 [2.3.3.4. Using the fips utility.](#)
 control panel : [6.7.3. The GNOME Control Center](#)
 Explorer : [6.5.3.10. File Manager](#)
 file/printer sharing : [10.3. Samba](#)
 filesystem : [4.3.4.4. Creating a directory.](#)
 gathering information via : [2.2.2. Collecting Configuration Information by Using Windows](#)
 MS-DOS prompt : [4.1.3. Issuing Commands](#)

notepad accessory : [4.5.2. Using the ae Editor](#)
running simultaneously with Linux : [8.2. Other Approaches to Desktop Computing](#)
as Samba server client : [10.4.1. Microsoft Windows Client](#)
shutting down : [4.1.1. Booting the System](#)
Start menu : [6.5.3.7. Pager](#)
taskbar : [6.5.3.4. Panel](#)

Windows applications

Microsoft Word : [1.1. Linux at Home and at Work](#)
running concurrently with Linux : [1.3. Reasons to Choose or Not Choose Linux](#)
WINE emulation and : [1.3. Reasons to Choose or Not Choose Linux](#)

Windows Explorer

[2.3.3.3. Shrinking a partition](#)
[2.3.3.4. Using the fips utility](#)
accessing Samba server via : [10.4.1. Microsoft Windows Client](#)

Windows MS-DOS prompt

[2.3.2. Viewing Partition Information](#)
[4.1.3. Issuing Commands](#)
[6.1.4. Terminal Windows](#)

Windows NT

[1.1. Linux at Home and at Work](#)
[1.2.1. PC Operating Systems](#)
[1.2.2. How Linux is Different](#)
[1.2.2.4. The Linux kernel](#)
as Samba server client : [10.4.1. Microsoft Windows Client](#)
version 4.0, compared : [1.2.3. Linux Features and Performance](#)

Windows partition, caution with : [8.2.2. WINE](#)

WINE ("WINE is not an emulator")

[1.3. Reasons to Choose or Not Choose Linux](#)
[8.2.2. WINE](#)
[9. Playing Linux Games](#)
[9.1. A Survey of Linux Games](#)
[9.2.3. Starcraft](#)
installing : [C.6.3.4. Installing the Mesa and WINE packages](#)

word processors

[1.1. Linux at Home and at Work](#)

[8.1.4. Other Desktop Applications](#)

StarWriter

[8.1.2. StarOffice](#)

[8.1.2.2. Running StarOffice](#)

Word (Applicxware) : [8.1.1.1. Running Applicxware](#)

WordPerfect for Linux (Corel) : [8.1.2.2. Running StarOffice](#)

wordcount : [13.2.6. Input/Output Redirection and Piping](#)

WordPerfect for Linux (Corel) : [8.1.3. Corel WordPerfect for Linux](#)
installing

[C.6.3.3. Installing WordPerfect for Linux](#)

[8.1.3.1. Installing WordPerfect for Linux](#)

running : [8.1.3.2. Running WordPerfect for Linux](#)

Words (Applicxware) : [8.1.1.1. Running Applicxware](#)

workgroup option (Samba server) : [10.3.2.1. Configuring global variables](#)
working directory

[4.3.3.1. Home and working directories](#)

[4.3.4.1. Displaying the working directory](#)

WS-FTP (Microsoft Windows) : [11.6. gFTP FTP client](#)

wvdial program

[11.1. Connecting to the Internet](#)

[11.3. Using wvdial](#)

[13.3.4. Periscope: A Useful Networking Script](#)

installing : [C.6.6.1. Installing the wvdial package](#)

Index: X

X Consortium

[1.2.2.5. The X Window System](#)

[5.1. What is X?](#)

X server, configuring with xf86config : [5.3. Configuring X](#)

X Window System

[1.2.2.5. The X Window System](#)

[5.1. What is X?](#)

caution when configuring : [5.2. Installing X](#)

installing : [5.2. Installing X](#)

starting : [5.4. Starting and Stopping X](#)

startup files : [6.5.1. Launching GNOME and Enlightenment](#)

stopping : [5.4. Starting and Stopping X](#)

switching to virtual consoles : [6.1.2. Using Virtual Consoles with X](#)

terminating : [6.1.3. Terminating X](#)

using : [6. Using the X Window System](#)

X servers : [C.6.1. Chapter 5: Installing and Configuring X Windows](#)

X-related packages : [C.6.1. Chapter 5: Installing and Configuring X Windows](#)

X11 interface : [10.4.3. Linux Client](#)

xf86config : [5.3. Configuring X](#)

XFree86 package : [5.1. What is X?](#)

XFree86 Project : [5.1. What is X?](#)

Index: Z

Z shell : [13.1.1. A Variety of Shells](#)

ZIP files : [4.3.5.7. Working with compressed files](#)

zsh shell : [7.1.6. Configuring Access to Shells](#)

[Symbols](#) | [A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Z](#)