The Most Complete Reference

Covers all three major Linux distributions

Extend your reach and capability with Apache

Special Edition

Covers Red Hat™, Slackware, & Caldera®

USING LINUX

Third Edition

Jack Tackett Jr. and David Gunter

Red Hat

Slackware

Caldera's Open Linux Lite

que®

What's HOT in This Edition!!

PW power!

Coverage of the three top Linux distributions: Red Hat, Slackware 96, and Caldera Open Linux Base

- Although this book focuses mainly on the Red Hat distribution, we still provide coverage on how to install and configure the Slackware 96 distribution.

- All three distributions are available on the accompanying CD-ROMs!

DEC Alpha coverage

- See how to install Red Hat Linux on various DEC Alpha platforms.

How to go online with Linux

- Full coverage of ftp, telnet, and other Internet programs is included.

- See how to create World Wide Web documents with HTML, as well as manage an Internet Web server.

- Three chapters have been added that explain how to get Apache, the world's most popular Linux Web server, up and running.

- The complete text to Que's *Running a Perfect Web Site with Apache* is included in HTML on CD-ROM!

How to go GUI with X Windows

- See how to install and configure XFree86, the most popular X Windows server for Linux.

- Learn how to use several exciting X applications.

- Find X Windows resource files on CD-ROM!

Special Edition

Using
Linux
Third Edition

que®

Special Edition

USING
LINUX
THIRD EDITION

Jack Tackett Jr. and David Gunter

Que®

Special Edition Using Linux, Third Edition

Credits

PRESIDENT
Roland Elgey

PUBLISHER
Stacy Hiquet

DIRECTOR OF MARKETING
Lynn E. Zingraf

PUBLISHING MANAGER
Fred Slone

SENIOR TITLE MANAGER
Bryan Gambrel

EDITORIAL SERVICES DIRECTOR
Elizabeth Keaffaber

ACQUISITIONS EDITOR
Tracy Dunkelberger

PRODUCT DIRECTOR
Rebecca J. Campbell

PRODUCTION EDITOR
Susan Shaw Dunn

EDITOR
Sherri Fugit

PRODUCT MARKETING MANAGER
Kristine Ankney

ASSISTANT PRODUCT MARKETING MANAGERS
Karen Hagen
Christy M. Miller

STRATEGIC MARKETING MANAGER
Barry Pruett

TECHNICAL EDITOR
Steven Burnett

TECHNICAL SUPPORT SPECIALIST
Nadeem Muhammed

SOFTWARE SPECIALIST
Brandon K. Penticuff

ACQUISITIONS COORDINATOR
Carmen Krikorian

SOFTWARE RELATIONS COORDINATOR
Susan D. Gallagher

EDITORIAL ASSISTANT
Andrea Duvall

BOOK DESIGNER
Ruth Harvey

COVER DESIGNER
Dan Armstrong

PRODUCTION TEAM
Jason R. Carr
Bryan Flores
Julie Geeting
Christy Hendershot
Sossity Smith

INDEXER
Ginny Bess

Composed in *Century Old Style* and *ITC Franklin Gothic* by Que Corporation.

To my wife, Peggy: I love and adore you.

—Jack Tackett Jr.

To my mother.

—David Gunter

About the Authors

Jack Tackett Jr. is a freelance writer and computer consultant in Research Triangle Park, N.C. He specializes in C/C++ and client/server development on various platforms. During his spare time, Jack enjoys traveling; writing; reading; and spending time with his friends, his family, and his family's two dogs and two cats in his new home in Cary, N.C. He invites your comments at **tackett@netwharf.com**.

David Gunter is an information technology consultant and author based in Cary, N.C. In addition to software development, he has been involved with supporting and managing diverse systems and networks for more than 10 years. David has a master's in computer science from the University of Tennessee. He has worked as both a lead and contributing author on 10 computer books, including the best selling *Special Edition Using Linux* series from Que. His publication credits include *Using Linux*, first, second, and third editions; *Using the Internet*, Third Edition; *Netscape Starter Kit*; *Using Netscape 3*; *Using Netscape 2*; *Client/Server Programming with RPC and DCE*; *Using UNIX*, Second Edition; and *Using Turbo C++ 4.5*. When not writing, consulting, researching, or surfing the Net, David spends as much time as possible with his wonderful wife, Lola, and their dog and two cats.

Contributing Authors

Lola Gunter is a technical consultant in Cary, N.C. She has worked in Web and multimedia software development as a computer consultant and as a manager of technical documentation. She has a bachelor's in computer science from the University of North Carolina-Asheville. Other than the Web and the Internet, her interests include working with stained glass, rough-housing with her German shepherd, and traveling.

Peggy Tackett is a software support specialist for Duke University. Webmaster for Duke University's Center for Aging and Human Development (**http://www.geri.duke.edu**), she teaches classes on using the Web. In her spare time, she enjoys gardening, working with dog-rescue groups, and traveling when she can pry her husband away from the computer.

Acknowledgments

We'd Like to Hear from You!

As part of our continuing effort to produce books of the highest possible quality, Que would like to hear your comments. To stay competitive, we *really* want you to let us know what you like or dislike most about this book or other Que products.

Please send your comments, ideas, and suggestions for improvement to

> The Expert User Team
> E-mail: **euteam@que.mcp.com**
> CompuServe: **72410,2077**
> Fax: (317) 581-4663

Our mailing address is

> Expert User Team
> Que Corporation
> 201 West 103rd Street
> Indianapolis, IN 46290-1097

You can also visit our team's home page on the World Wide Web at

> **http://www.mcp.com/que/developer_expert**

Your comments will help us to continue publishing the best books available in today's market.

Thank you,

The Expert User Team

Contents at a Glance

VI | Using the Internet

VII | Setting Up a Web Site

Appendixes

Table of Contents

6 Installing the X Windows System   159

III | Managing the File System

14 Managing File Systems 261

15 Understanding the File and Directory System 287

IV | Working with Linux

VI | Using the Internet

29   Using Electronic Mail   569

VII | Setting Up a Web Site

Appendixes

Introduction

What's new in the third edition of *Special Edition Using Linux*? Lots! First, we've revamped the book to highlight the Red Hat distribution—probably the most popular and easiest-to-install Linux distribution ever. You'll also find another CD-ROM containing a *lite* version of Caldera's Open Linux Base product. But we didn't forget Paul Volkerding's venerable Slackware distribution, so we also provide installation and usage information for Slackware 96. So you get not one, not two, but *three* complete Linux distributions for the price of a single book!

Next, we added an entire section of chapters on getting Apache—the world's most popular Web server software—up and running under Linux. In fact, on the accompanying CD-ROM you'll find the complete text to Que's *Running a Perfect Web Site with Apache*. As for the rest of the book, you'll find updates and new material throughout the chapters as we try to bring you the best introduction to Linux in the world.

However, if you're just tuning in, you might want to know just what the heck Linux is.

In 1991, Linus Torvalds, then a 23-year-old college student, began a personal project to expand the Minix operating system into a full-fledged clone of the UNIX operating system so popular on college campuses. The project is still evolving—Linux is continuously updated and expanded by literally hundreds of people spread around the world.

Well, Linux is a unique animal in the computer (r)evolution. It isn't a commercial product backed by a huge corporation; rather, it's an operating system born of frustration and built by a ragtag team of computer enthusiasts around the world. This team used Internet resources to communicate and build the operating system named Linux.

But don't think Linux is just a hobby for hackers around the world—it's not! Plenty of commercial products are being written specifically for Linux. In fact, several companies are porting their UNIX-based applications to Linux, such as Corel's WordPerfect. You'll also find many Fortune 500 companies using Linux for internal projects and mission-critical applications. Not bad for a product not controlled by any one organization.

You'll also find plenty of free applications and utilities for Linux. Since Linux's inception, almost the entire GNU library of utilities has been ported to Linux, and the X Windows GUI system— so popular on UNIX-type workstations—also has been ported. GNU (a recursive acronym for GNU's Not UNIX) is a project started by one man to make software available to anyone who wants access. The GNU General Public License in Appendix D describes the philosophy under which Linux and many other fine software packages are distributed. The accompanying CD-ROMs contain many of these packages.

This book provides you with enough information to use and enjoy Linux. The accompanying CD-ROMs contain the Red Hat 4.0 distribution and the Slackware 96 distribution, each of which use the 2.0.18 Linux kernel.

Probably the first order of business is to help you pronounce the word *Linux*. To most Americans, the pronunciation is LEN-nucks, with the short i sound. The official pronunciation is LIE-nucks, with the long i sound. ▪

ON THE WEB

You can see and hear Linus pronounce Linux at the URL **http://www.cse.psu.edu/~drost/linus.html**.

N O T E   If you don't understand what a Uniform Resource Locator (URL) is or how to use it, don't despair! This book will help you learn your way around the Internet by using Linux. ▪

▶ **See** "Understanding URLs," **p. 557**

Who Should Use This Book?

Anyone interested in the Linux phenomenon can use this book as a guide to installing, configuring, and using Linux. Linux is often called a UNIX clone, but it's actually a POSIX-compliant multiuser, multitasking operating system for Intel 386 and later processors. POSIX is an international standard for operating systems and software detailing interoperability standards. Linux doesn't require MS-DOS or Windows to operate; in fact, Linux can replace those programs on your computer.

Because Linux is still evolving, it's imperative that you understand the possibility of losing existing data on your system. *Do not install Linux without first backing up your system*. It might

be necessary to repartition your hard drive to make room for this new operating system, although it's possible to install Linux on top of MS-DOS or to repartition your hard drive without losing data. This book and the task of installing Linux aren't for the novice user. With the proper precautions, however, anyone can install and enjoy Linux.

N O T E The most current version of Linux is always available on the Internet, from sources listed in Appendix A, "Sources of Information." The accompanying CD-ROMs contain the latest possible versions of Linux, but due to the rapid development of this popular operating system and the chaotic process in which it's developed, it's impossible to provide the latest and greatest on a CD-ROM. In fact, although all efforts are made to keep the book and CD-ROMs in synch, that also is nearly impossible. Unlike commercial software, which changes infrequently and under controlled conditions, Linux and related software are perpetually dynamic.

Because Linux is very close to UNIX, many of the operations and procedures necessary for using Linux also apply to many UNIX systems. By learning to use Linux, you also learn how to use most UNIX systems.

UNIX has evolved over the years to become the premier operating system used by hundreds of thousands of people throughout the world. This isn't an accident. Earlier versions of UNIX were harder to manipulate than other operating systems, but despite this, UNIX managed to amass a distinguished following in academic and scientific circles. These professionals realized not only what a powerful, flexible, and manageable operating system UNIX was, but also its potential to be the best operating system ever. Their efforts have culminated in the UNIX of today, with its marvelous utilities, bundled with the newest communications capabilities and graphical user interfaces (GUIs).

The UNIX of today promises again to revolutionize the personal computer industry, and perhaps redirect the industry's growth. UNIX has evolved from a minicomputer operating system to one that crosses all hardware platforms. There's no reason to think that this evolution will stop. UNIX may well become the standard for what most users dream of—complete standardization and compatibility of all computer systems eventually, regardless of size or power.

UNIX comes in several flavors from a variety of vendors, including versions for the Intel PC platforms, but most of these versions cost big bucks. Linux provides a relatively inexpensive—free if you have access to the Internet—solution to learning about UNIX-type procedures and commands, the X Windows GUI, and accessing the Internet via Linux.

Who Should *Not* Use This Book?

If you are a Linux kernel hacker or a UNIX guru, this book may not be your cup of tea. This book is a great resource for someone wanting to know more about Linux and UNIX who has never been involved with either operating system.

However, if you know how to install Linux and maneuver around in UNIX, you may still find this book of use, particularly if you are only a UNIX user and have never had the chance to perform system administration tasks. Several sections of the book explain the finer points of

system administration and how to maintain a Linux/UNIX system. Typically, a normal UNIX user is never allowed to perform these system administration tasks, but with Linux you become king of the hill and ruler of the system, free to do whatever you want to do!

Now, if you don't have a clue what MS-DOS is or what a floppy disk looks like, you might want to brush up on some computer basics before tackling Linux. Linux isn't for the faint of heart—you must have some understanding of how a computer works. If the thought of repartitioning or reformatting your hard drive sends shivers down your spine, you probably should put off learning Linux for a while until you become more comfortable with your computer system.

Hardware Needed to Use This Book

Because Linux isn't a commercial software product, it hasn't gone through the rigorous quality assurance tests that most commercial software products endure. Also, most of Linux has been written across the Internet by computer *hackers* (not *crackers*, but people who truly enjoy writing software that accomplishes something)—thus, the hardware supported by Linux is the hardware owned by the various hackers. Table 1 is a brief list of the supported hardware. If you don't have the correct hardware, it's unlikely that you'll be able to boot Linux and productively use the system—forewarned is forearmed! Appendix C, "The Linux Hardware Compatibility HOWTO," provides a more in-depth listing of Linux-supported hardware.

Table 1   A Brief List of Hardware Supported by Linux

Item	Description
CPU	Intel 386 and later (and compatibles), DEC Alpha, Sun Sparcs, and PowerMacs.
Bus	ISA, EISA, VESA local bus, and PCI; the MicroChannel bus isn't fully supported yet.
RAM	Minimum of 2M of RAM; 4M is recommended.
Hard drive	AT standard hard drive controller; Linux supports controller MFM, RLL, ESDI, and IDE controllers. Linux also supports several popular SCSI drive and CD-ROM drive controllers.
Disk space	Minimum of 20M; 80M is recommended.
Monitor	Linux supports Hercules, CGA, EGA, VGA, and SVGA video cards and systems; X Windows has other requirements detailed in Chapter 6, "Installing the X Windows System."
Mouse	Any standard serial mouse (for example, Logitech, Microsoft, or Mouse Systems) or bus mouse from Microsoft, Logitech, or ATIXL.
CD-ROM drive	Any CD-ROM drive that uses a true SCSI interface works; some proprietary CD-ROM drives such as the SoundBlaster series are

Item	Description
	also supported. CD-ROM drives known to work with Linux include NEC CDR-74, Sony CDU-45, Sony CDU-31a, Mitsumi CD-ROMs, and Texel DM-3042.
Tape drive	Any SCSI tape drive works; other drives hosted from a floppy controller may also be supported. Now, the Colorado Jumbo 120 and 250 using the QIC 80 format are supported.
Printer	If you can access your parallel printer from MS-DOS, you should be able to access it from Linux; some fancy features might not be accessible.
Ethernet card	If you have access to an Ethernet network, Linux supports several standard Ethernet cards for accessing your network. Cards supported include 3Com's 3C503, 3C509, and 3C503/16; Novell's NE1000 and NE2000; and Western Digital's WD8003 and WD8013.

ON THE WEB

The Web site **http://glycerine.itsmm.uni.edu/mca/** provides more information on MicroChannel bus support.

How to Use This Book

You may prefer to read this book from cover to cover. The information progresses from simple to complex as you read through the various sections and their chapters. Because the information is separated into seven parts and four appendixes, each with its own particular emphasis, you can choose to read only those parts that appeal to your immediate needs. Don't, however, let your immediate needs deter you from eventually giving attention to each chapter. Whenever you have the time, you can find a wealth of information in them all!

Part I: Installing Linux
Part I, "Installing Linux," provides a detailed overview of the Linux system as well as instructions to get Linux up and running. It consists of seven chapters:

- Chapter 1, "Understanding Linux," introduces the Linux operating system.
- Chapter 2, "Overview of Features," provides a general overview of the various components that make up the Linux system and the various distributions. The chapter also covers some of the new commercial software available for Linux.
- Chapter 3, "Installing Red Hat," gives detailed instructions for installing the version of Red Hat provided on the accompanying CD-ROM.
- Chapter 4, "Installing Slackware 96," gives detailed instructions for installing the version of Slackware provided on the accompanying CD-ROM.

- Chapter 5, "Running Linux Applications," provides a basic introduction to the process of setting up your Linux system.
- Chapter 6, "Installing the X Windows System," provides you with the necessary information to get the X Windows system up and running under Linux. Under Linux, the X Windows system is called XFree86 and is similar to other GUI environments, such as Microsoft Windows or the OS/2 Workplace Shell.
- Chapter 7, "Using X Windows," provides you with information necessary to use the X Windows system under Linux.

Part II: Systems Administration

Part II, "Systems Administration," provides basic information on configuring and managing a typical Linux installation. It consists of six chapters:

- Chapter 8, "Understanding Systems Administration," provides a brief background of the processes and procedures needed to configure and maintain a Linux system.
- Chapter 9, "Booting and Shutting Down," details the various actions that happen when you boot up or shut down a Linux system, and explains why you can't simply switch off the power supply.
- Chapter 10, "Managing User Accounts," shows you how to add, delete, and manage user accounts on your machine.
- Chapter 11, "Backing Up Data," explains the necessity of backing up your data, as well as the procedures needed to back up your Linux system.
- Chapter 12, "Improving System Security," gives you a brief overview of system security on Linux systems and then explains the procedures needed to maintain a reasonably secure system.
- Chapter 13, "Upgrading and Installing Software," provides you with the information needed to install new software from the Internet, and also tells you how to patch existing programs.

Part III: Managing the File System

Part III, "Managing the File System," provides detailed knowledge of how to be more productive with various Linux features. Everything you learn in these three chapters can be transferred easily to other UNIX-type systems:

- Chapter 14, "Managing File Systems," provides an overview of creating, mounting, and using a file system under Linux.
- Chapter 15, "Understanding the File and Directory System," looks at subtle nuances of Linux command syntax, execution from the command prompt, parsing, command interpretation, and command feedback.
- Chapter 16, "Managing Files and Directories," details the Linux file system structure and organization, file-naming conventions, and directory hierarchy. The chapter also teaches you how to successfully navigate the Linux file system.

Part IV: Working with Linux

Part IV, "Working with Linux," increases your skill at working with the Linux command-line tools and utilities. Part IV consists of five chapters:

- Chapter 17, "Understanding Linux Shells," introduces you to the magical world of Linux shells, the powerful capabilities that exist through the use of shell scripting, and the different shells you may encounter with different versions of Linux.

- Chapter 18, "Managing Multiple Processes," explores the capabilities of Linux when you run more than one process at a time. You learn how to initiate and manage multiple processes, as well as how to control and stop them.

- Chapter 19, "Using the *vi* Editor," shows you how to use UNIX's visual editor. Although vi isn't the most productive editor in the world, every Linux/UNIX system has it, and sometimes it's the only editor available for use.

- Chapter 20, "Using the *emacs* Editor," teaches you how to use the ubiquitous UNIX editor written by GNU patriarch Richard Stallman.

- Chapter 21, "Printing," covers all the printing basics, from issuing print commands and checking printer status to canceling print jobs and dealing with common printing problems.

Part V: Network Administration

Part V, "Network Administration," provides a greater understanding of the procedures and processes necessary to administer a robust Linux system. Four chapters make up Part V:

- Chapter 22, "Understanding the TCP/IP Protocol Suite," provides an overview of the most common network transport protocol suite in use today on the Internet.

- Chapter 23, "Configuring a TCP/IP Network," shows you how to set up and configure TCP/IP on Linux.

- Chapter 24, "Configuring Domain Name Service," provides you with the necessary information to get your system up and running with Domain Name Service (DNS).

- Chapter 25, "Using SLIP and PPP," illustrates how to configure and use Serial Line Internet Protocol (SLIP) and Point-to-Point Protocol (PPP) lines to connect with the Internet.

Part VI: Using the Internet

The five chapters in Part VI, "Using the Internet," provide a basic overview of the Internet:

- Chapter 26, "Understanding the Internet," explains this network of networks that has become the information superhighway.

- Chapter 27, "Accessing the Network with *telnet, ftp*, and the *r-* Commands," provides you with information on how to use various programs such as telnet and ftp to access information around the world.

- Chapter 28, "Surfing the Internet with the World Wide Web," gives you an overview of using various Linux utilities to search for and retrieve information from the Internet, with emphasis on the Web.

- Chapter 29, "Using Electronic Mail," gives you an overview of electronic mail (e-mail) and how to use it in Linux.
- Chapter 30, "Surviving Usenet news," provides you with an explanation of Usenet newsgroups, and then provides you with instructions for accessing this global community of newsgroups.

Part VII: Setting Up a Web Site

Part VII, "Setting Up a Web Site," provides detailed information on setting up and running a Web server on Linux. Five chapters make up this part:

- Chapter 31, "Creating Web Documents with HTML," tells you how to create home pages for the World Wide Web by using HTML for your Linux system.
- Chapter 32, "Getting Started with Apache," gives you the basics so that you can get started with Apache. It covers compiling and installing Apache, and the basic configuration options.
- Chapter 33, "Configuring Apache," discusses the major configuration options within Apache, including MIME types, indexing, server-side includes, image maps, and virtual hosts.
- Chapter 34, "Managing an Intranet Web Server," discusses the various administrative concepts involved with managing a Web server, including controlling server child process, increasing server efficiency, managing log files, and dealing with security issues.

Appendixes

Appendix A, "Sources of Information," provides you with a detailed listing of books, magazines, Usenet newsgroups, and FTP sites dealing with Linux. Also, you get a brief glimpse of the myriad resources available to you as a Linux user.

Appendix B, "The Linux HOWTO Index," provides a list of all the main and mini HOWTOs available. HOWTOs provide information on *how to* accomplish a specific task with Linux. This HOWTO comes right off the Internet.

Appendix C, "The Linux Hardware Compatibility HOWTO," provides important details on the hardware supported by the current Linux distribution. This HOWTO also comes right off the Internet.

Appendix D, "The GNU General Public License," is the verbatim license for using GNU applications. It describes your responsibilities when using GNU programs.

Appendix E, "The Caldera Open Lite CD-ROM," provides a brief introduction to one of the most popular commercial products based on Linux—Caldera Open Linux Base.

Conventions Used in This Book

This book uses several special conventions that you need to become familiar with. These conventions are listed here for your reference.

Linux is a *case-sensitive* operating system; that means that when this book instructs you to type something at a command or shell prompt, you must type exactly what appears in the book, exactly as it is capitalized. This book uses a `monospaced typeface` for Linux commands to set them off from standard text. If you're instructed to type something, what you are to type appears in `monospaced bold`. For example, the book instructs you to do the following:

Enter `cat`.

You must press the letters <c>, <a>, and <t>, and then press the <Return> key.

Keys are sometimes pressed in combination; when this is the case, the keys are presented like this: <Ctrl-h>. This example implies that you must press and hold the <Ctrl> key, press the <h> key, and then release both keys.

N O T E This book uses a convention for key names that may vary from what you are accustomed to. To avoid confusion in the case-sensitive UNIX environment, this book uses lowercase letters to refer to keys when uppercase letters may be the norm. For example, this book uses the form <Ctrl-c> instead of the form <Ctrl-C> (the latter form may make some readers wonder whether they should press <Ctrl> and <Shift> and <c>).

Some example listings show a portion of the screen after you type a specific command. These listings show the command prompt or shell prompt—usually a dollar sign ($)—followed by what you type in `bold`. Don't type the dollar sign when you follow the example on your own system. Consider this example:

```
$ lp report.txt &
3146
$
```

You should type only what appears in `bold` on the first line (that is, type `lp report.txt &` and then press <Return>). The rest of the listing shows Linux's response to the command.

When discussing the syntax of a Linux command, this book uses some special formatting to distinguish between the required portions and the variable portions. Consider the following example:

```
lp filename
```

In this syntax, the `filename` portion of the command is a variable; that is, it changes depending on what file you actually want the `lp` command to work with. The `lp` is required because it's the actual command name. Variable information is presented in *italic*; information that must be typed exactly is not in italic.

In some cases, command information is optional—that is, it's not required for the command to work. Square brackets ([ ]) enclose optional parts of the command syntax. Consider the following example:

```
lp filename [device1] [abc]
```

Here, `lp` is the command name and is neither optional nor variable. The *device1* parameter is both variable and optional (it is in italic and enclosed in square brackets); this means that you can type any device name in place of *device1* (without the brackets), or type nothing at all for that parameter. The abc parameter is optional (you don't have to use it if you don't want to), but it's not variable; if you use it, you must type it exactly as it appears in the book—again, without the brackets.

Tips, notes, and cautions appear throughout the book in special formats to make the information they contain easy to locate. Longer discussions not integral to the flow of the chapter are set aside as *sidebars*, with their own heading.

The book also contains many cross-references to appropriate topics throughout the book. A typical cross-reference appears as follows:

▶ **See** "Using X Windows," **p. xxx**

Installing Linux

Understanding Linux

To understand Linux, you must first understand the question, "What is UNIX?" The reason is that Linux is a project initiated to create a working version of UNIX on Intel-based machines, more commonly referred to as IBM PC-compatible computers that most people are familiar with.

UNIX is arguably the most versatile and popular operating system found today on scientific and high-end worksta-tions. This chapter explains why you may want to select the UNIX-like Linux instead of one of the other operating systems available for Intel platforms, such as MS-DOS, Windows 95, Windows NT, or OS/2. ∎

What is Linux?

Linux is a freely accessible operat-ing system.

Why use Linux?

This chapter briefly touches on the many uses you may have for Linux.

What are the different versions of UNIX?

UNIX comes in many flavors, each having a common base and—sometimes—ancestry. This chapter examines the history of this fasci-nating OS.

What are the features of Linux/UNIX?

This chapter provides a brief over-view of the many features available to the user.

Who owns Linux?

This chapter discusses the oftentimes confused issue of who owns Linux.

What Is Linux?

Linux is an operating system for several types of computer platforms, but primarily Intel-based PCs. The system has been designed and built by hundreds of programmers scattered around the world. The goal has been to create a UNIX clone, free of any commercially copyrighted software, that the entire world can use.

Actually, Linux started out as a hobby of Linus Torvalds while a student at the University of Helsinki in Finland. He wanted to create a replacement for the Minix operating system, a UNIX-like operating system available for Intel-based PCs.

N O T E If some of the terms used in this chapter are unfamiliar to you, don't worry because we'll explain many of them within the chapter. ■

Why Use Linux?

If you have a computer, you must have an operating system. An *operating system* is a complex set of computer codes that provides the operating process *protocols*, or laws of behavior. Without an operating system, your computer sits idly, incapable of interpreting and acting on your input commands or running a simple program. Most operating systems are commercial programs supported by major software companies. If you buy an operating system, you must be content with what that vendor offers. Because the program is a proprietary program, you can't modify or experiment with the operating system internals.

Application programs are software packages you purchase to perform certain activities, such as word processing. Each package is written for a specific operating system and machine. Again, you have little control on how modifications are made to the application program and when or if updates are ever performed.

You'll want to use Linux because it's the only operating system today that's freely available to provide multitasking, multiprocessing capabilities for multiple users on IBM PC-compatible hardware platforms. No other operating system gives you these same features with the power that Linux enjoys. Linux also separates you from the marketing whims of the various commercial providers. You aren't locked into upgrading every few years and paying outrageous sums to update all your applications. Many applications for Linux are freely available on the Internet, just as the source code to Linux itself is available on the Internet. Thus, you have access to the source code to modify and expand the operating system to your needs—something you can't do with commercial operating systems such as Windows NT, Windows 95, MS-DOS, or OS/2.

Freedom from commercial vendors is also a potential downside to using Linux. Because no one commercial vendor supports Linux, getting help isn't just a phone call away. Linux can be finicky and may or may not run properly on a wide range of hardware. The potential to damage or delete data files now residing on your system also exists because Linux is constantly changing and doesn't go through a rigorous testing process before being released.

Linux isn't a toy; it's a system designed to give users the feeling of tinkering with a new project, just like in the beginning of the PC revolution. However, Linux is relatively stable on many systems and thus provides you an inexpensive opportunity to learn and use one of the most popular operating systems in the world today—UNIX. Many CD-ROM vendors and software companies, such as Red Hat and Caldera, now support the Linux operating system. Linux is an alternative to other UNIX systems and can be used in place of those sometimes expensive systems. Thus, if you program on UNIX systems during the day, you can have a UNIX-like system at home. Are you a systems administrator of a UNIX system at work? You can perform some of your duties from home by using Linux. Don't have a clue as to what UNIX is? Well, then, Linux provides a low-cost introduction to one of the most popular operating systems in the world—UNIX.

Linux also provides you with easy access to the Internet and the rest of the information super-highway.

UNIX Versions

The name *UNIX* can mean many things. It's one of the most context-specific terms used in the computer industry today. To better understand what's meant when someone uses the word UNIX, you must know which version of the operating system is being referred to.

Now UNIX is a trademark of X/Open and must be licensed to be sold. Although versions of UNIX are offered by many different vendors, these versions fall into one of two major categories: System V or Berkeley Software Distribution (BSD). UNIX System V is the version developed by the creators of UNIX, AT&T Bell Laboratories. BSD UNIX is associated with the version of UNIX that was developed and proliferated by the University of California at Berkeley. The most popular version of BSD UNIX was sold by Sun Microsystems for its powerful work-stations.

N O T E With UNIX System V Release 4, AT&T and Sun Microsystems merged their two versions of UNIX into UNIX System V. Because of this merger and the rallying of the industry around the UNIX System V standard, UNIX System V is now considered the standard UNIX system. ∎

Undoubtedly, the best thing to happen to the UNIX operating system was the development of System V, Release 4.2 (SVR4.2). Created and introduced by UNIX System Laboratories (USL, now owned by Nippon Santa Cruz, Inc., a wholly owned subsidiary of Santa Cruz Operation, Inc.), SVR4.2 combines the power and capabilities of UNIX with a "user-affectionate" graphical user interface (GUI). The result is an operating system that can be easily navigated by computer novices or radically customized by system administrators and other experts.

Before the release of SVR4.2, UNIX users saw only a command-line prompt (a simple dollar sign) after logging into the system. With no on-screen guidance evident, inexperienced users frequently were intimidated into forsaking UNIX in favor of other, easier-to-use operating systems. No doubt this (often) intimidating dollar sign prompt contributed to the UNIX reputation as an unfriendly operating system throughout its early years. Although experience and word of mouth dispelled some user reluctance, it was a reputation dying a slow death—until now.

An Overview of UNIX/Linux Features

The benefits derived from using the UNIX operating system, and hence Linux, stem from its power and flexibility. These are the result of the many features built into the system, ready for you to use as soon as you turn the system on. The following sections examine these features more closely.

Multitasking

The word *multitasking* describes the capability to seemingly execute multiple programs at the same time without hindering each application from running. Most UNIX variants provide a type of multitasking called *preemptive multitasking*. The name is based on the fact that each program is guaranteed a chance to run; each program executes until the operating system pre-empts it to allow another program to run. This type of multitasking is exactly what Linux does.

MS-DOS and Windows 3.1 don't support preemptive multitasking, but they do support a form of multitasking called *cooperative multitasking*. Under cooperative multitasking, programs run until they either voluntarily allow another program to run, or have nothing else to do for the time being.

To better understand Linux's multitasking capability, examine it from another view. The microprocessor in your computer can do only one thing at a time, but it can complete those individual tasks in periods of time so short they're difficult to comprehend. For example, typical microprocessors today operate at clock speeds of 75 to 300 MHz (megahertz) and faster. This means that they can transfer from 75 to 300 and more million bits per second. When processing a complete set of instructions, the speeds are much faster, typically 6 nanoseconds (billionths of a second). The human mind can't detect the difference between so short a delay and something occurring simultaneously. In short, it appears that tasks are performed at the same time.

Using a real-world example may further explain multitasking. Suppose that, while working in a spreadsheet application, you discover that the current month's information hasn't yet been merged from the database to the spreadsheet application. Typically, this takes some minutes to accomplish, so you start the process. Then you switch to another terminal window in Linux to work on a report you're preparing. Before you finish the report, you switch back to the original Linux virtual terminal to check on the progress of your spreadsheet merge process. Seeing that it's not yet complete, you switch back to the report terminal, only to realize that some of the information you want to include is located in a letter you wrote some time ago. So you open yet another window, select the file containing your letter, extract the information you need by using a copy utility, and return to the report, where you paste the copied information into place—all the while using the same word processor. Linux informs you that your merge process is now complete, so you close your word processing windows and return to the original spreadsheet window to finish your work there.

You can readily see the benefits of having preemptive multitasking capabilities. In addition to reducing *dead time* (time where you can't continue working on an application because a

process hasn't yet finished), the flexibility of not having to close application windows before opening and working in others is infinitely more convenient.

Linux and other preemptive multitasking operating systems accomplish the preemptive processing by monitoring processes waiting to run, as well as those now running. The system then schedules each process to have equal access to the microprocessor. The result is that open applications appear to be running concurrently; in reality, there are delays of only billionths of a second between when the processor executes a set of instructions from one application and when it's scheduled by Linux to devote time to that process again.

It's the capability to allocate time to running applications from within a freely available operating system that sets Linux apart from other operating systems and environments available today, such as Windows 3.1 and Windows 95, MS-DOS, and commercial versions of UNIX. The other big feature of Linux is that you have access to the source code for the operating system. Thus, if you want to change how the system works and have the necessary capability to implement the change, you can modify the Linux operating system to your heart's content. Commercial vendors will never allow you access to their source code to modify their wares.

Multiuser

The concept of many users accessing applications or processing power from a single PC was only dreamed of a few short years ago. UNIX and Windows NT helped develop that dream into reality. Linux's capability of allocating microprocessor time to many applications at once naturally lent itself to serving many people at once, each running one or more applications. The truly remarkable feature of Linux and its multiuser, multitasking features is that more than one person can work in the same version of the same application at the same time, from the same or separate terminals. Don't confuse this with multiple users updating the same file simultaneously, a feature that's potentially confusing, potentially hazardous, and positively undesirable.

Again, a real-world scenario helps further your understanding of the multiuser feature. Imagine a company whose personnel are networked to a Linux system. While one employee places the finishing touches on outgoing correspondence, another produces a company roster. Still another uses the same word processing package to prepare a view-graph presentation by importing pictures from a graphics package; meanwhile, the boss extracts information from a database for a report he's preparing, using (you guessed it) the same word processing application.

Down the hall in accounting, three data-entry clerks are updating files in the same accounting database; in the mailroom, the supervisor updates the shipping file that simultaneously updates the accounting database. At the same time, the system administrator sets up an account for a new employee and deactivates a second system account for an employee who is away on an extended leave of absence.

Although it may be difficult to imagine this scenario, it's being repeated thousands of times each day in offices throughout the world that use the UNIX operating system. Linux is a freely available UNIX clone, and many systems are configured for scores of people to access; others are set up with only a few user accounts.

Programmable Shells

The *programmable shell* is another feature that makes UNIX, and hence Linux, what it is: the most flexible operating system available. Within the shell's framework is a whole new world available to those adventurous enough to master Linux's command syntax nuances.

Although you shouldn't be discouraged from attempting exciting forays into the Bourne, C, or bash shells, you should understand that, without proper guidance and preparation, shell programming can be a difficult lesson in frustration. Make sure that you have a reliable reference manual at hand (such as the one you're reading now) and that you have the phone number of a good system administrator or other Linux expert, in case you get bogged down.

▶ **See** " Understanding Shells," **p. 323**

For those of you who anticipate the rewarding experience of directing a powerful operating system such as Linux, the importance of command syntax in Linux can't be overstated. (*Syntax* is nothing more than the order and form of a command line.) The Linux shell scans each command line to determine whether the formation and spelling are consistent with its protocols.

The shell's scan process is termed *parsing*—the commands are broken down into more easily processed components. Each component is interpreted and executed, including those special characters that convey additional meaning to the shell. These special characters are further expanded into their proper command processes and executed.

Although many UNIX and Linux versions include more than one type of shell, they all function in essentially the same way. A shell works as the interpreter between the user and the kernel (the heart, or brain, of the Linux operating system). The primary difference among the three available shells lies in command-line syntax. Although it's not a severe limitation, you'll encounter difficulties if you try to use C shell commands or syntax in the Bourne or bash shells.

Perhaps the simplest of scenarios to help further your understanding of UNIX and Linux shell programming is that of *background processing*. For example, you should back up all or parts of your system files at one time or another for protection against accidental erasure or hardware problems. If those files and their associated directories are of substantial size, the backup process can take a considerable amount of time.

With a simple shell program consisting of a line or two of command arguments, a backup process can run concurrently with other operations you may want to do. After programming the shell and starting the backup process, you can simply open a different window to access the desired application and begin your work there. When the background processing is completed, UNIX signals that the job is finished.

Linux shell programming serves as many different functions as there are people willing to attempt it. Many use this programming feature to personalize their systems and make them more user-friendly. Others find it helpful in streamlining many of the applications they run by allowing them to perform a number of processes in the background so that they're free to work in others.

Some users go even further by devising programs that link processes or applications so that they can reduce their workload to perhaps a single session of data entry and have the system

update numerous software packages at once. Only the user's imagination limits what can be done with Linux shell programming.

Device Independence Under UNIX

At first glance, whether your computer system's peripherals can operate on a stand-alone, or independent, basis, may not seem important. When you view these peripherals from a multiuser UNIX environment, however, device independence becomes fundamental to a productive workplace. To understand the importance of device independence, you must first understand how other systems view attached peripherals and how UNIX views them.

Until recently, computer systems could generally support peripherals such as printers, terminals, disk drives, and modems. The technology explosion has added other devices too numerous to list. Difficulties are encountered when the user can't utilize a peripheral because an operating system can't access the device. This incapability can be the result of incompatible system architecture, operating system addressing limitations, and so on.

UNIX sidesteps the problem of adding new devices by viewing each peripheral as a separate file. As new devices are required, the system administrator adds the required link to the kernel. This link, also known as the *device driver*, ensures that the kernel and the device merge in the same fashion each time the device is called on for service.

As new and better peripherals are developed and delivered to users, the UNIX operating system allows immediate, unrestricted access to their services as soon as the devices are linked to the kernel. The key to device independence lies in the adaptability of the kernel. Other operating systems allow only a certain number or a certain kind of device. UNIX can accommodate any number of any kind of devices because each device is viewed independently through its dedicated link with the kernel.

Device Independence Under Linux

Linux shares many of the same benefits of device independence as UNIX. Unfortunately, one of Linux's best features—freedom from a commercial entity—is also one of its biggest liabilities. Linux has been developed over the last few years by various programmers from around the world. These programmers don't have access to all the equipment created for the IBM PC and compatibles. In fact, Linux can't run on IBM PCs that support the MicroChannel bus. Linux does support a wide variety of PC hardware, as illustrated in Appendix C, "The Linux Hardware Compatibility HOWTO."

However, because Linux is a UNIX clone, it too has an adaptable kernel; as more programmers join the Linux project, more hardware devices are added to the various Linux kernels and distributions. As a last resort, because you have the source code to the kernel, you, or someone you hire, can modify the kernel to work with new devices.

Communications and Networking

UNIX's superiority over other operating systems is equally evident in its communications and networking utilities. Linux is no exception. No other operating system includes such tightly

coupled networking capabilities, and no other operating system has the built-in flexibility of these same features. Whether you need to talk with another user through an e-mail utility or download large files from another system across the nation, Linux provides the means to do so.

Accessing the Internet is a hot topic today, and people are spending hundreds of dollars on software so that their PCs can hitch a ride on the information superhighway. With Linux, you get all that software for free! The Internet was created and thrives in the UNIX world and, thus, Linux is quite at home there. With Linux, you can communicate almost instantly with the rest of the world.

Next, you can accomplish internal messaging or file transfer through a number of Linux commands, including write, call, mail, and mailx, as well as cu and uucp. As an aid to understanding these capabilities, it's helpful to understand that information exchanged by users on the same system (also called *internal communications*) is accomplished by terminal-to-terminal communication, e-mail, and an automatic calendar that serves as a scheduler/information manager.

An extension of Linux communications is found in the cu and uucp commands (used for external communications). Not only does Linux allow for file and program transfer, it gives system administrators and technicians a window of access to other systems. Through this remote-access capability, one technician can effectively service many systems, even when those systems are located across great distances.

Networking in UNIX needs little explanation. The communications capabilities inherent in the operating system were designed to support many tasks and many users over many miles. UNIX quite naturally evolved in the professional marketplace as the operating system of choice, thanks to the same characteristics that gave it premier status in the scientific community and academia. Linux is set to follow that same route as a unique replacement for commercial UNIX systems.

Open Systems Portability

In the never-ending quest for standardization, many organizations have taken a renewed interest in the direction in which operating systems are developing. UNIX hasn't gone unnoticed. The drive to standardize UNIX stems from the many UNIX variants now available. You learn more about how those variants were developed in the following section.

Efforts have been made to combine, collate, and otherwise absorb all versions of UNIX into a single, all-encompassing version of the operating system. Initially, the effort met with guarded enthusiasm, and some effort was expended on coming to terms with blending the different versions. As with many noble efforts, this one was doomed to failure because developers weren't willing to sacrifice part of what they had already invested into their particular version. (Sad to say, many developers still feel that way.)

However, the continued existence of UNIX varieties isn't necessarily cause for alarm. Despite the different varieties, all are still inherently superior to all other operating systems available today because each contains the same elements described in the preceding pages.

Portability is merely the capability to transport an operating system from one platform to another so that it still performs the way it should. UNIX is indeed a portable operating system. Initially, UNIX could operate on only one specific platform—the DEC PDP-7 minicomputer. Today, the many UNIX variants can operate in any environment and on any platform, from laptops to mainframes.

Portability provides the means for different computer platforms running UNIX to communicate accurately and effectively with any of the other platforms without the addition of special, high-priced, after-market communications interfaces. No other operating system in existence can make this claim.

A Brief History of Linux

The history of Linux is tied to the history of UNIX and, to a lesser extent, a program called Minix. Minix was an operating system tutorial written by the well-known and respected computer scientist Andrew Tannebaum. This operating system became popular on several PC platforms, including MS-DOS-based PCs. But more on Minix later. First, a brief history of UNIX.

Although AT&T created the UNIX operating system, many other companies and individuals have tried to improve the basic idea over the years. The following sections examine a few of the leading variants in use today.

AT&T

Ken Thompson, a computer programmer for AT&T Bell Laboratories, and a group of people working under Ken's direction developed an operating system that was flexible and completely compatible with programmers' varied needs. Legend tells how Ken, who had been using the MULTICS operating system, dubbed this new product UNIX as he joked with others on his development team. He was lampooning the MULTICS multiuser operating system: UNIX was derived from *uni*, meaning *one* or *single*, followed by the homophone X. Perhaps the greater joke in this bit of folklore lies in the fact that MULTICS is remembered by few users today as a viable multiuser operating system, while UNIX has become the de facto industry standard for multiuser, multitasking operating systems.

BSD

Berkeley Software Distribution (BSD), University of California at Berkeley, released its first version of UNIX, based on AT&T's Version 7, in 1978. BSD UNIX, as it's known throughout the industry, contained enhancements developed by the academic community at Berkeley designed to make UNIX more user-friendly. The user-friendly "improvements" in BSD UNIX were an attempt to make UNIX appeal to casual users, as well as to advanced programmers who liked its flexibility in conforming to their changing demands. Despite being less than 100 percent compatible with AT&T's original UNIX, BSD did accomplish its goals: The added features enticed casual users to use UNIX.

BSD has become the academic UNIX standard. The original creators of BSD have since released a version for the Intel platform called, appropriately enough, BSD. This version too has a limited distribution on the Internet and via CD-ROM vendors. The authors also wrote several articles a few years ago in the computer magazine *Dr. Dobb's Journal* detailing the design and implementation of BSD386 or FreeBSD. Today BSDI, the commercial version of FreeBSD, is another popular operating system similar to Linux.

USL

UNIX System Laboratories (USL) was an AT&T spinoff company that had been developing the UNIX operating system since the early 1980s. Before Novell purchased it in 1993, USL produced the source code for all UNIX System V derivatives in the industry. However, USL itself didn't sell a shrink-wrapped product at that time.

USL's last release of UNIX was UNIX System V Release 4.2 (SVR4.2). SVR4.2 marked USL's first entry into the off-the-shelf UNIX marketplace. In a joint venture with Novell, which temporarily created a company called Univel, USL produced a shrink-wrapped version of SVR4.2 called UnixWare. With Novell's purchase of USL, Novell had shifted the focus of USL from source-code producer to UnixWare producer. Novell has now sold its version of UNIX to the Santa Cruz Operation (SCO).

Recently, SCO made a free single-user license available to the public for using SCO UNIX. The program, however, costs $19 for the distribution media, not unlike Linux. However, whereas SCO provides a copy of its operating system, it doesn't provide the source code. Some in the Linux community suspect Linux is giving the UNIX community—or at least the SCO community—some stiff competition.

XENIX, SunOS, and AIX

Microsoft developed its UNIX version, XENIX, in the late 1970s and early 1980s, during the peak of the PC revolution. Processing power available in PCs began to rival that of existing minicomputers. With the advent of Intel's 80386 microprocessor, it soon became evident that XENIX, which had been developed specifically for PCs, was no longer necessary. Microsoft and AT&T merged XENIX and UNIX into a single operating system called System V/386 Release 3.2, which can operate on practically any common hardware configuration. XENIX is still available today from Santa Cruz Operation (SCO), a codeveloper with Microsoft, whose efforts to promote XENIX in the PC market have made this version of UNIX one of the most commercially successful.

Sun Microsystems has contributed greatly to UNIX marketability by promoting SunOS and its associated workstations. Sun's work with UNIX produced a version based on BSD. Interestingly enough, AT&T's SVR4 is compatible with BSD, too—no doubt an offshoot of AT&T and Sun Microsystems' collaboration in UNIX System V Release 4.0.

IBM's venture into the world of UNIX yielded a product called AIX (Advanced Interactive Executive). Although AIX isn't as well known as some other UNIX versions, AIX performs well and has no problem holding its share of the operating system market. It's perhaps the old

mindset that any UNIX version is an unfriendly, unforgiving operating system that has kept AIX from a better market reception.

Linux

Linux is the brainchild of a computer science student named Linus Torvalds. Linux began life as a hobby project in 1991 for Linus, then 23. He hoped to create a more robust version of UNIX for Minix users. Minix, as stated earlier, is a program developed by computer science professor Andrew Tannebaum.

The Minix system was written to demonstrate several computer science concepts found in operating systems. Torvalds incorporated these concepts into a stand-alone system that mimics UNIX. The program was widely available to computer science students all over the world and soon generated a wide following, including its own Usenet newsgroups. Linus Torvalds set out to provide a better platform for his fellow Minix users that could run on the widely available IBM PC. Linus targeted the emerging 386-based computers because of the task-switching properties of the 80386 protected-mode interface.

What follows are some of the announcements Linus made when announcing his Linux program.

N O T E   These announcements are from the "Linux Installation and Getting Started Guide," by Matt Welsh (copyright 1992-94 by Matt Welsh, 205 Gray Street NE, Wilson, NC 27893, **mdw@sunsite.unc.edu**). They're used subject to section 3 of Matt's copyright.

You can obtain the complete "Linux Installation and Getting Started Guide" from the Linux Documentation Project's various archives sites. You can find this book on **sunsite.unc.edu** in the directory **/pub/ Linux/docs/LDP/install-guide.** For information on how to access archives and download files, refer to Chapter 28, "Surfing the Internet with the World Wide Web."

> "After that it was plain sailing: hairy coding still, but I had some devices, and debugging was easier. I started using C at this stage, and it certainly speeds up development. This is also when I started to get serious about my megalomaniac ideas to make 'a better Minix than Minix.' I was hoping I'd be able to recompile gcc under Linux some day….

> "Two months for basic setup, but then only slightly longer until I had a disk driver (seriously buggy, but it happened to work on my machine) and a small file system. That was about when I made 0.01 available [around late August of 1991]: it wasn't pretty, it had no floppy driver, and it couldn't do much [of] anything. I don't think anybody ever compiled that version. But by then I was hooked, and didn't want to stop until I could chuck out Minix."

In a later announcement, made in **comp.os.minix** on Oct. 5, 1991, Linus introduced to the world Linux version 0.02, the first official version of Linux.

> "Do you pine for the nice days of Minix 1.1, when men were men and wrote their own device drivers? Are you without a nice project and just dying to cut your teeth on an OS you can try to modify for your needs? Are you finding it frustrating when everything

works on Minix? No more all-nighters to get a nifty program working? Then this post might be just for you."

"As I mentioned a month ago, I'm working on a free version of a Minix lookalike for AT-386 computers. It has finally reached the stage where it's even usable (though may not be depending on what you want), and I am willing to put out the sources for wider distribution. It's just version 0.02, but I've successfully run bash, gcc, gnu-make, gnu-sed, compress, and so forth under it."

Who Owns Linux?

IBM owns the rights to OS/2, Microsoft owns the rights to MS-DOS and MS Windows, but who owns the rights to Linux? First and foremost, Linux isn't public domain software; various components of Linux are copyrighted by many people. Linus Torvalds holds the copyright to the basic Linux kernel. Red Hat, Inc. owns the rights to the Red Hat distribution version, and Paul Volkerding holds the copyright to the Slackware distribution. Many Linux utilities are under the GNU General Public License (GPL). In fact, Linus and most Linux contributors have placed their work under the protection of the GNU GPL. You can find the license on each of the accompanying CD-ROMs in the root directory in the file named copying.

This license is sometimes referred to as the GNU Copyleft (a play on the word *copyright*). This license covers all the software produced by GNU (itself a play on words—GNU's Not UNIX) and the Free Software Foundation. The license allows programmers to create software for everyone. The basic premise behind GNU is that software should be available to everyone, and that if someone wants to modify the program to his or her own ends, that should be possible. The only caveat is that the modified code can't be restricted—others must have the right to the new code.

The GNU Copyleft, or GPL, allows a program's creators to keep their legal copyright, but allows others to take, modify, and sell the resulting new program. However, in doing so, the original programmers can't restrict any of these same rights to modify the program from the people buying the software. If you sell the program as is or in a modified form, you must provide the source code. That's why Linux comes with the complete source code.

From Here...

Linux is a viable alternative to UNIX on the desktop. The freely available source code and applications make Linux a reasonable alternative to other operating systems for PC-based platforms. For more information, check out the following:

- Chapter 3, "Installing Red Hat," provides information on putting the Red Hat distribution of Linux on your computer.

- Chapter 4, "Installing Slackware 96," provides information on putting Paul Volkerding's Slackware 96 distribution of Linux on your computer.

- Chapter 5, "Running Linux Applications," explains how to use some of the applications that come on the accompanying CD-ROMs.

Installing Red Hat

This chapter gives you the information you need to install the Red Hat distribution of Linux. Remember, Linux isn't a commercial product, and you might find some problems. Although this book leads the way, you may find the need to use the resources, such as the various HOWTOs, provided on the Red Hat CD-ROM. However, Red Hat is one of the easiest distributions to install, so take heart! Although there are many advanced methods to installing Linux, this chapter focuses on installing from the CD-ROM. ▨

The hardware you need to install Linux

Linux doesn't support all hardware components in all configurations. Check to make sure you have the necessary hardware.

How to prepare your system for the installation

Installing Linux requires some preparation, like formatting your hard drive. You must also plan the layout of your disk space for Linux's optimum performance.

How to partition your hard drives to make room for Linux

Linux requires certain partitions to work effectively, especially if swap space is required. You must also decide whether other operating systems will coexist with Linux and prepare for their installation as well.

How to install Red Hat Linux

After you prepare your system for installation, you must create your installation disks, select your installation media (the accompanying Red Hat CD-ROM), and begin the installation.

What to do if things go wrong

Because Linux has no formal technical support, you need to know what to do if you have problems with the installation.

N O T E This book assumes that you have a working knowledge of DOS and of such items as formatting your hard drive, partition tables, and sector sizes. If this information sounds like a foreign language, check out *Using MS-DOS 6.2,* Special Edition, or ask a computer guru buddy to help you through this.

T I P You're about to make major changes to your system, so be careful. It's a good idea to have paper and pen nearby to take notes just in case something does go wrong; besides, you'll need to jot down some numbers along the way.

Understanding Linux's Hardware Requirements

To be able to install Linux successfully, you need supported hardware. Choosing the right level of hardware for your Linux system depends on such factors as the number of users to be supported and the types of applications to be run. All this translates into requirements for working memory, hard disk storage space, the types of terminals needed, and so forth.

ON THE WEB

For current information on hardware that Red Hat 4.1 supports and doesn't support, see Red Hat's Web site at **http://www.redhat.com/support/docs/rhl-intel/rh40-hardware-intel.html**.

Most Linux systems today consist of PCs. These Linux installations are often for only a single user, although they may also be tied into larger Linux or UNIX systems.

CAUTION

Linux is a constantly evolving system, and hardware support is sporadically updated. The Red Hat distribution contained on the accompanying CD-ROM is relatively stable, but new hardware support may have been provided by the time this book is printed and the CD-ROM is created. Although many hardware components have clones or compatible replacements, not all may work with Linux. If you do have the hardware discussed in this chapter, the odds are excellent that Linux installs, boots, and operates properly. If you don't have the equipment listed, Linux may or may not operate properly, but the odds are against your getting a system up and running.

If you're using a version of Linux in a single-user configuration (the most likely configuration), you're the system administrator. It's your responsibility to understand the system well enough to perform the administrative duties required to keep it operating at an optimum level. These duties include keeping enough space on the hard drive, backing up regularly, ensuring that all devices attached to the system have the proper software drivers, installing and configuring software, and so forth.

Choosing the level of hardware you need depends heavily on the hardware used by the myriad people who programmed the Linux system. Unlike commercial software developers who can afford to test their systems on many different hardware configurations, Linux developers typically have access only to their personal computers. Luckily, because so many Linux developers exist, most of the standard hardware found in the PC world is supported.

The System's CPU

A basic system requires an IBM-compatible PC with an Intel 80386 or later CPU in any of the various CPU types, such as the 80386SX, 80486DX/2, and Intel's various Pentium processors. Other CPU clones, such as the 80386 clone chips made by Cyrix and Advanced Micro Devices (AMD), are also compatible with Linux.

The 80386 and 80486SX processors don't have math coprocessors built in, but Linux doesn't require a floating-point math coprocessor. Linux can emulate the coprocessor by using software routines, but at a significant reduction in execution speed. For a fast system, you should consider getting a CPU with a math coprocessor built in, such as an 80486DX or Pentium.

Part

I

Ch

3

The System's Bus

The type of bus used to communicate with the peripherals is also important. Linux works only with the ISA and EISA buses. The MicroChannel Architecture (MCA) bus used on IBM's PS/2 isn't supported, although a port is in process. Some newer systems use a faster bus for such items as disk access and video displays, called the *local bus*. Linux does support the VESA Local Bus but may not support a non-VESA Local Bus architecture. Linux also supports the PCI (Peripheral Component Interconnect) bus found on many high-end Pentium systems.

Memory Needs

Linux requires surprisingly little RAM to run, especially when compared to comparable operating systems such as OS/2 and Windows NT. Linux requires at least 2M of RAM, although 4M is highly recommended. If you have less than 4M of RAM, you need to use what's called a *swap file*. The basic rule of thumb is that the more memory your system contains, the faster your system runs.

The next memory consideration for Linux is the use of the X Windows clone called XFree86. XFree86 is a version of X Windows that's freely distributable and is included with Linux for that reason. XFree86 is a GUI similar to Microsoft Windows.

▶ **See** "Installing the XFree86 System," **p. 163**

To productively use XFree86, your Linux system requires at least 16M of virtual memory. *Virtual memory* is the combination of physical memory and swap space on the hard drive. Again, the more physical memory contained on the system, the more responsive the system, especially when using XFree86.

Disk Drives and Space Requirements

Although it's possible to run Linux from a floppy-drive-only system, running Linux from your system's floppy drive isn't recommended.

> **N O T E** You can boot Linux from a floppy drive. *Booting a system* refers to the process of starting a computer system and loading the operating system into memory to start the system. The term is derived from the phrase *bootstrapping*. ▪

For a home-based system, you need a floppy drive—either 5 1/4 inch or 3 1/2 inch. You need a floppy drive even if you install and run Linux from your CD-ROM.

For better system performance, you want to install Linux on a hard drive. You must have an IBM AT standard drive controller. This shouldn't be a problem because most modern, non-SCSI controllers are AT compatible. Linux supports all MFM and IDE controllers, as well as most RLL and ESDI controllers. Linux may or may not support the newer, high-capacity IDE drives from older 8-bit IDE controllers.

Linux supports a wide range of SCSI hard-drive controllers. If your controller is a true SCSI controller—that is, not a proprietary version of SCSI—Linux can use your controller. Linux now supports SCSI controllers from Adaptec, Future Domain, Seagate, UltraStor, the SCSI adapter on the ProAudio Spectrum 16 card, and Western Digital. The following card types are supported:

Adaptec 152x/1542/1740/274x/284x/294X	Always IN2000
Buslogic	Pro Audio Spectrum 16
EATA-DMA (DPT, NEC, AT&T)	Qlogic
Seagate ST-02	Trantor T128/T128F/T228
Future Domain TMC-8xx, 16xx	UltraStor
Generic NCR5380	7000FASST
NCR 53c7, 8xx	

▸ **See** "Controllers (IO)," **p. 706**

If you have the proper drive controller, you must worry about disk space requirements. Linux supports multiple hard drives and can be installed across drives. Unlike other operating systems, Linux doesn't need to be installed on the same hard drive; pieces can be installed on different drives.

Disk Space   The one item to realize about disk space, however, is that Linux can't reside on the same partition as other operating systems, such as MS-DOS and OS/2. *Partitions* are areas of a disk drive specified during initialization of a drive and before formatting a drive. You typically use a program called fdisk to partition a drive. Some commercial products allow you to

repartition a drive, and Linux provides a utility called FIPS to do the same. To use Linux efficiently, you should repartition your hard drive and allocate enough space for the Linux system files and your data files.

> **CAUTION**
>
> Unless you're installing Linux onto a brand-new hard drive, you'll need to repartition and reformat the drive. This destroys all information now stored on that portion of the drive. Thus, it's imperative to back up your files—twice—before installing Linux. If space permits, you can split a single hard drive into multiple partitions and copy your files back to one of the partitions.

The amount of disk space required depends on the software you install and the amount of data you expect that software will generate. Linux requires less disk space than most implementations of UNIX systems. You can run a completely functional Linux system, without X Windows support, in 20M. For a complete installation of everything in the distribution, 150M to 200M is recommended.

Swap Space Finally, as mentioned earlier in the section "Memory Needs," if you have limited RAM, you need swap space. Whereas systems such as Microsoft Windows create a swap file that resides on your hard drive as any other file, Linux allows the swap file to reside on a separate swap partition. Most Linux installations use partitions rather than files. Because you can place multiple partitions on the same physical hard drive, you can place the swap partition on the same drive as Linux, but for better performance you should place the swap partition on a separate drive.

Linux allows up to eight swap partitions that can be no larger than 16M. A rule of thumb is to set the swap file size to twice the amount of physical RAM contained on your system. Thus, if you have 8M of physical RAM, your swap partition should be 16M in size.

Monitor Requirements

For text-based terminals, Linux supports all standard Hercules, CGA, EGA, VGA, and SuperVGA video cards and monitors. To take advantage of the color-coding directory listings available with Linux, you need a color monitor. So for text-based operation, any video/controller combination should work.

The big problems occur when you run the X Windows system distributed with Linux. To use XFree86, you need a video adapter that uses one of the chipsets listed in Table 3.1. *Chipsets* are a group of integrated circuits, or computer chips, used to take information from the computer and convert the data into a form displayable on a video monitor. To find out the chipset used by your video adapter, check the documentation included with your card to determine whether there are any problems using XFree86.

Part
I

Ch
3

Table 3.1   Video Chipsets Supported by Linux

Manufacturer	Chipset(s)
Tseng	ET3000, ET40000AX, ET4000/W32
Western Digital	WD90C00, WD90C10, WD90C11, WD90C24, WD90C30, WD90C31
Trident	TVGA8800CS, TVGA8900B, TVGA8900C, TVGA8900CL, TVGA9000, TVGA9000i, TVGA9100B, TVGA9200CX, TVGA9320, TVGA9400CX, TVGA9420
ATI	28800-4, 28800-5, 28800-a
NCR	77C22, 77C22E, 77C22E+
Cirrus Logic	CLGD5420, CLGD5422, CLGD5424, CLGD5426, CLGD5428, CLGD6205, CLGD6215, CLGD6225, CLGD6235
OAK	OTI067, OTI077
S3	86C911, 86C924, 86C801, 86C805, 86C805i, 86C928
Compaq	AVGA
Western Digital/ Paradise	PVGA1

N O T E   The release notes for the current version of XFree86 distributed with Linux should contain a more recent list of supported and non-supported chipsets.

Some problems encountered by the XFree86 developers are caused by adapter manufacturers who don't provide the necessary information on programming the cards to display information. Without this information, developers can't support X Windows on those adapters. Also, some manufacturers provide this information but require either royalty payments or non-disclosure agreements for others to use the information. These types of restrictions make it impossible to support these adapters on a freely distributed system like the XFree86 system for Linux.

N O T E   The Diamond video cards haven't been supported in the past, due to restrictions the company had on giving out proprietary information. The Diamond company has now begun work with the XFree team to support the company's video systems specifically under Linux and XFree86.

▶ **See** "Video Cards," **p. 700**

CD-ROMs

To install the Linux system included on the accompanying CD-ROM, you must have a CD-ROM drive supported by Linux. Because most CD-ROMs use a SCSI interface controller, any SCSI controller listed earlier in the section "Disk Drives and Space Requirements" should also

work with a CD-ROM attached to the controller. Linux also now supports many of the new EIDE and ATAPI CD-ROMs available on the market.

Many of the CD-ROMs included with multimedia packages may or may not work with Linux, depending on whether the controller is a true SCSI adapter or a proprietary adapter. Proprietary adapters for the most part won't work. However, Linux does specifically support the Creative Labs SoundBlaster line of CD-ROMs and provides a specific installation configuration for their CD-ROMs. Other CD-ROMs known to work with Linux are as follows:

NEC CDR-74	Okano
Sony CDU-541	Wearnes CD with interface card
Sony CDU-31a or 33a	SoundBlaster, Panasonic Kotobuki, Matsushita, TEAC-55a, or Lasermate
Plextor DM-3024	Most IDE/ATAPI CD-ROMs
Aztech	Mitsumi CD-ROMs
Orchid	

▶ **See** "CD-ROM Drives Supported," **p. 712**

Network Access

You can connect a Linux system to the world in several ways, but the two most popular (and supported) methods are via network controller cards and modems. Network controller cards include Token Ring, FDDI, TAXI, and Ethernet cards. Most common business networks use an Ethernet controller card.

Network Access via Ethernet   Ethernet, a protocol invented by Xerox, has gained immense popularity in the networking world. Although it's unlikely you'll connect Linux to an Ethernet network at home, many business and educational institutions are connected via Ethernet. Table 3.2 lists several of the Ethernet adapters supported.

Table 3.2   Ethernet Controller Cards Supported by Linux

Manufacturer	Interface Card
3Com	3c503, 3c503/16, 3c509
Novell	NE1000, NE2000
Western Digital	WD8003, WD8013
Hewlett-Packard	HP27245, HP27247, HP27250

Part

I

Ch

3

Network Access via Modem   At home, you'll more than likely connect to the outside world via a modem and a communications protocol such as SLIP or PPP. Linux supports almost every type of modem on the market, internal and external. If you can access the modem from MS-DOS, you'll have no problem accessing the modem from Linux.

▶ **See** "Understanding the Requirements for SLIP and PPP," **p. 506**

Miscellaneous Hardware

The following sections list miscellaneous hardware supported by Linux, such as mice, tape drives, and printers. Although such hardware makes Linux easier to use and more robust, it isn't required.

Mice   Using text-based Linux doesn't require a mouse. Unlike many UNIX implementations, however, Linux does allow you to cut text from any area of the screen and paste it to the command line by using a mouse. If you intend to use the X Windows clone, XFree86, you must use a mouse.

Linux supports most serial mice, including the following mice:

- Logitech
- MM series
- Mouseman
- Microsoft
- Mouse Systems

Linux also supports the Microsoft, Logitech, ATIXL, and PS/2 bus mice. In fact, any pointing devices, such as trackballs and touch screens, which emulate the previously listed mice, should work with Linux.

Tape Drives   Tape drives provide a great deal of storage space for backing up your computer system. Linux supports several SCSI-based tape systems, as shown in Table 3.3. Linux also supports the popular Colorado Memory Systems tape drives (120 and 250 versions), which are plugged into a system's floppy-disk controller. The versions that plug into the printer port aren't now supported. Most drives supporting QIC-02 should also work with Linux.

Table 3.3   Tape-Backup Drives Supported by Linux

Manufacturer	Model
Exabyte	All SCSI models
Sanko	CP150SE
Tandberg	3600
Wangtek	5525ES, 5150ES, 5099EN

▶ **See** "Tape Drives—Supported," **p. 712**

Printers   Linux supports the complete range of parallel printers. Configuring Linux to support serial printers is tedious and error-prone. Serial printer support isn't well documented or supported by the basic Linux installation programs. If you have a serial printer, you may have problems using it under Linux. If you have a parallel printer, your biggest problem is most likely the *stair-step effect*:

```
This is line one.
             This is line two.
                           This is line three.
```

How UNIX, and hence Linux, treat carriage returns and line feeds produces the stair-step effect. Under most UNIX systems, the command to move the paper down one line (line feed) and then position the print head at the beginning of the line (carriage return) are represented by one control character. Under systems such as MS-DOS and Windows, however, each command is represented by a different control character. When you print a UNIX file under a printer configured for MS-DOS systems, you see the stair-step effect because the file contains only the line-feed control character and not the carriage-return control character.

▶ **See** "Knowing What You Need to Configure Printers," **p. 438**

Part

I

Ch

3

Starting the Installation Process

To start the installation process, you need one or two (depending on the installation method) 3 1/2-inch 1.44M formatted floppy disks. These disks will be used to create a boot disk for the Linux installation.

Next, you should make sure that you have enough hard disk space to install Linux. Everything on the CD-ROM, if installed, requires about 300M of disk space, but you can get by with less, especially if you don't install the X Windows system. To decide on the amount of space, you should decide how much space you want for user accounts—that is, the space you want to provide to your users. On a single-user system, 30M is more than enough.

Next, decide how much swap space your machine needs. If your machine has 8M of RAM or less, you need a 24M swap space. If you have 16M or more, your swap space should equal the amount of RAM.

Finally, figure about 30M for your root directory. This is the main directory from which all other directories under Linux are accessed.

▶ **See** "Looking at Linux Standard Directories," **p. 297**

Again, a minimal installation should fit in 200M, whereas a full installation, with plenty of user space, should fit well on a 500M drive.

N O T E   You can also run part of the Linux file system from the CD-ROM without installing all the software. You can choose to do so during installation. ■

If you decide to install and configure X (highly recommended), you should also write down what type of chipset your video card uses. If you have a serial mouse and modem, write down the serial port that each is using. You need this information later during the configuration process.

Understanding the Various Installation Methods

We, the authors, envision most of you installing Red Hat from the accompanying CD-ROM. However, you can use one of the following four methods to install Red Hat: from CD-ROM, via NFS, via FTP, or from a hard drive.

To install directly from CD-ROM, you need access to DOS. From the DOS prompt, execute the command

```
[cdrom-drive]:\dosutils\autoboot
```

where [cdrom-drive] is the drive letter for your system's CD-ROM.

> **CAUTION**
>
> This method will erase your hard drive. Back up any files you fear losing.

If you have another partition available, you can install Linux to coexist with your system without erasing what's already there. To do this, you need the CD-ROM, an empty partition, and a boot disk. You learn later in this chapter how to create the boot disk, as well as how to repartition your hard drive.

NFS (Network File System) provides a way to install Red Hat across a network. First, you must mount the CD-ROM drive on a machine supporting the ISO-9660 file system with RockRidge extensions, and then export the file system via NFS. You need to know the path to the exported file system and the IP number, or, if DNS is configured, the name of the system.

FTP (File Transfer Protocol) is a method for transferring files across the Internet. (Chapter 27, "Accessing the Network with *telnet*, *ftp*, and the *r*– Commands," explains FTP in more detail.) To install via FTP requires a boot disk and the supplemental disk described later in this chapter.

Installing Red Hat from a hard drive requires the same boot and supplemental disks used for an FTP installation. First, create a directory named RedHat. Then copy the corresponding directory from the CD-ROM, and all the subdirectories there, to the RedHat directory. You can use the following DOS commands to do this:

```
cd \RedHat
xcopy /s e:\RedHat
```

The cd command assumes that you're already on the installation hard drive; the xcopy command assumes that your CD-ROM drive is drive E.

No matter which method you use, you'll need at least the boot disk to proceed with installation. But first, you should gather some information.

Compiling Needed Information

Before starting the installation, you need the following information about your system:

- The type of video card, chipset, and monitor used
- The serial port used by your mouse
- The serial port used by your modem
- The network information for your computer, if it's connected to a network (items such as its IP address, gateway, and domain name)
- The type of hard drive and CD-ROM drive in your system, and their controller type
- The name you intend to call your system

If you're connecting to the Internet, you can get most of this information from either your network administrator or from your Internet service provider.

If you intend to use other operating systems on the same computer (such as Windows 95, Windows NT, or OS/2), you'll need to create the necessary partitions for these operating systems. Typically, you need to use the operating system's partitioning software, because Linux can't handle these other partition types.

Part
I

Ch
3

ON THE WEB

A product named System Commander, from V Communications, lets you install and switch between 32 different operating systems. You can find more information about this product at **http://www.v-com.com/**.

Next, you should check for any last-minute changes to the Red Hat distribution. The reasons are many, but the two major reasons are that Linux is constantly updated, and this chapter is being written at least a month before the CD-ROM is cut. In the interim, new material or bug fixes may have been released. You can contact InfoMagic, the manufacturers of the accompanying CD-ROM, for more information.

ON THE WEB

You can also check for updated material on the Web at **http://www.redhat.com/errata**.

If you're not installing directly off the CD-ROM, you next need to repartition your current hard drive to make room for Linux. This may cause problems, because repartitioning a hard drive destroys any data contained on the affected partitions. After making room for Linux, you need to boot the Linux system and create its new partitions and file systems. Typically, Linux systems need a primary partition to store the files on and a swap file partition, especially if you have a machine with 8M or less of memory.

NOTE A file system is basically a section of your hard drive specially formatted to hold certain types of files. UNIX and Linux use file systems to represent entire sections of the directory tree. This is in contrast to MS-DOS, which places subdirectories in the directory tree on the same logical drive. UNIX systems use the directory tree format because placing subdirectories on different drives is safer. If one drive malfunctions, only the information on that drive needs to be replaced or fixed.

▶ See "Understanding File Systems," p. 262 ▪

After creating the file systems, you then install the Linux operating system, its support files, and various application packages distributed with the system. To install Linux, you must first boot a stripped-down version of the operating system. You do this by creating a boot disk and a supplemental disk set containing the stripped-down OS.

Creating the Boot and Supplemental Disks

You need to create the boot and supplemental disks by using the rawrite program. You can find this program on the accompanying CD-ROM in the /dosutils subdirectory. For this step, you need two formatted floppies: one labeled *boot*, the other labeled *supp*. Place the boot disk in the A drive and enter the following:

```
E:\dosutils>rawrite

Enter disk image source file name: e:\images\boot.img
Enter target diskette drive: A:
Please insert a formatted diskette into drive A: and press -ENTER-
```

If you want to abort the process, simply press <Ctrl-c> to stop. If rawrite fails, try a new formatted disk. If the problem persists, you should have your hardware checked for possible problems.

After writing the boot disk, you need to create the supp disk. Simply use the supp image file name (supp.img) as the source file name in the preceding command sequence.

Partitioning Your Hard Drive

After you back up your system and make the necessary boot and supplemental disks, you must prepare your system's hard drive for Linux.

CAUTION
This process is the most dangerous because maximum data loss is assured. If you haven't backed up your system, do so now. Although you can use an experimental program called FIPS and commercial programs such as Partition Magic to do non-destructive repartitioning, a full backup is recommended, just in case problems occur.

Understanding Partitions

In the early days of PCs, hard drives were few and far between. Most computers used floppies to hold the operating system, programs, and their data. With the introduction of the IBM PC XT, IBM introduced a 10M hard drive. Early operating systems such as DOS could access only a limited amount of space on hard drives. Then hard drive manufacturers kept expanding the space on their hard drives more quickly than the operating system's capability to access the additional space. The operating system got around this problem by letting the user split the hard drive into sections, called *partitions*. These partitions can hold program files, other operating systems, or data.

Typical MS-DOS systems have one partition, which is referred to as drive C. If you split the drive into partitions, these partitions are typically referred to in alphabetical order as D, E, and so on. MS-DOS also allows you to install multiple hard drives, so that the next drive in this chain might be referred to as F.

UNIX and Linux don't use drive letters to refer to partitions; instead, they use directory names to refer to partitions. Also, as indicated earlier, Linux users can place different directories on different partitions and even on different drives. You can also place different operating systems on different partitions.

Partitions are specified in a section of the hard drive referred to as the *boot record* in what's called a *partition table*. This table is used by the various operating systems to determine what operating system to boot and where their files can physically be found on the hard drive. The boot record is used to *boot*, or start up, the machine's operating system. Linux's boot program, LILO (which stands for LInux LOader), and other boot managers use this section of the hard drive, typically found on the first sectors of the drive, to control which operating system to start.

The partition table holds information about the locations and sizes of the various partitions on the hard drive. There are three kinds of partitions: primary, extended, and logical. DOS and some other operating systems must boot from primary partitions. Hard drives can contain only four primary partitions. An extended partition doesn't contain data itself; instead, it allows the user to define other, logical partitions on the drive. Thus, to get around the limit of four primary partitions, you can define an extended partition and then define other logical partitions within the extended partition.

Some operating systems, such as MS-DOS and versions of OS/2 before version 2.0, require that they be installed in a primary partition, but they can access logical drives in extended partitions. This is important to remember if you're going to have a DOS system and a Linux system reside on the same drive. DOS must go in a primary partition.

Using *FDISK*

Partitions are created, destroyed, and managed by a program usually called FDISK. Each operating system has its own version of FDISK, so be sure to use the correct one. If you're now

using DOS or are planning to use DOS, you must first repartition the DOS drive by using DOS's FDISK. You later use the Linux version of fdisk to create the Linux partitions. If you're using OS/2, you also need to use the OS/2 version of FDISK to prepare the OS/2 partitions.

Partition Requirements   First, you should plan what partitions you need. DOS requires a primary partition; Linux and OS/2 can reside in other partitions. If you're using the OS/2 boot manager, which also works well with Linux, you must prepare for its use. You must also be aware that if you're shrinking a current DOS partition to make room for Linux, not all of your files can be restored to the new, smaller DOS partition.

N O T E   You can access DOS partitions from Linux, thus moving, saving, and editing DOS files under Linux. But you can't execute DOS programs under Linux.

Two experimental components of Linux allow you to emulate DOS under Linux and also install Linux under DOS. Both systems are still in the implementation stage and are, thus, more suited for Linux hackers. Also, one of the methods, UMSDOS, is incompatible with Red Hat, so you can't use this method at all. You can find plenty of information on these topics in the Linux world.

▶ **See** "Running DOS Programs Under Linux," **p. 139**

Next, you should jot down the number of partitions you need and how much disk space to provide each with.

DOS Requirements   If you want to boot DOS, it must go in a primary partition. A bootable version of DOS doesn't require much space—just enough for the system files, COMMAND.COM, CONFIG.SYS, and any driver files needed to start your system. For example, I provide a 5M DOS partition on my first drive to boot DOS.

When DOS is loaded and running, you can access any of the other extended and logical drives on the system. Unfortunately, whereas Linux can access DOS files in a DOS partition, DOS can't access Linux files in a Linux partition.

OS/2 Requirements   OS/2 versions 2.0 and later don't need a primary partition. The OS/2 system can install and boot from an extended partition. Thus, you can install DOS on a primary partition and create an extended partition area for OS/2 and Linux. The space required for OS/2 is version- and feature-dependent; you should consult your OS/2 documentation for space requirements. You should also subtract 1M from available space if you intend to use the OS/2 boot manager.

Linux Requirements   As explained earlier, Linux stores files on file systems, and these file systems can reside on different partitions, basically as safety precautions. Linux requires one partition for each file system. The next consideration is for a swap partition. Linux, like most operating systems that use disk space for memory (called a *virtual memory configuration*), needs a swap file or a swap partition to simulate physical memory using disk space. Linux typically uses a swap partition.

The size of the swap partition depends on the amount of physical RAM your system contains. A rule of thumb is to make your swap partition twice the size of your amount of RAM. Thus, if

you have 8M of RAM in your system, you should create a swap partition 16M in size. If you have 4M of RAM or less, you must activate a swap partition.

Linux swap partitions can be only 128M in size, so if you need more space, you have to create multiple swap partitions. Thus, if you have a Linux system that needs two partitions for Linux (one for the system files and one for user files), plus a swap partition, you need to define two Linux partitions and one 32M swap partition.

Repartitioning Your DOS Drive

This section assumes that you need to repartition a DOS drive. First, you execute FDISK by typing fdisk at the DOS prompt. The FDISK Options screen appears (see Figure 3.1).

FIG. 3.1
From the FDISK Options screen you can look at current partitions, create new partitions, and delete old partitions.


```
                         MS-DOS Version 6
                       Fixed Disk Setup Program
                 (C)Copyright Microsoft Corp. 1983 - 1993

                            FDISK Options

Current fixed disk drive: 1

Choose one of the following:

1. Create DOS partition or Logical DOS Drive
2. Set active partition
3. Delete partition or Logical DOS Drive
4. Display partition information
5. Change current fixed disk drive


Enter choice: [1]




Press Esc to exit FDISK
```

The screen shown in Figure 3.1 might look different depending on which version of MS-DOS you're using. Pick menu option 4, Display Partition Information. The Display Partition Information screen appears (see Figure 3.2). Write down the information in this screen. You need the current partition table information if you decide to abort the Linux installation and put your system back the way it was before you started.

FIG. 3.2
You can look at current partition information by using the Display Partition Information screen in MS-DOS 6.x


```
                          Display Partition Information

Current fixed disk drive: 1

Partition  Status   Type    Volume Label  Mbytes   System   Usage
C: 1         A    PRI DOS   OPUS_DOS         5      FAT12      4%
   2              Non-DOS                    8                 6%
   3              Non-DOS                  376               100%
   4              Non-DOS                  114                90%

Total disk space is  127 Mbytes (1 Mbyte = 1048576 bytes)




Press Esc to continue
```

An Alternative to Repartitioning Your Hard Drive

You may not need to repartition your hard drive, although it's thought that repartitioning offers the best introduction to Linux. You can use FIPS to non-destructively repartition your hard drive.

FIPS stands for *First non-destructive Interactive Partition Splitting*. A program developed by Arno Schaefer as a result of the Linux project, FIPS is used to move around DOS partitions to make room for Linux partitions.

You can find the complete instructions for using FIPS in the document fips.doc located on the accompanying Red Hat CD-ROM in the /utils/fips directory. This program can help only if you have enough free space left on your drive to install Linux; otherwise, you need to delete unneeded files or use the process described earlier to repartition your hard drive.

Under Slackware Linux (whose installation is discussed in Chapter 4), you can install Linux on the same partition as DOS with UMSDOS. UMSDOS is a project to allow Linux to exist on DOS partitions. UMSDOS lets you create the Linux root file system under an existing DOS directory. Unfortunately, you can't use UMSDOS under Red Hat.

Deleting Partitions   Unfortunately, FDISK doesn't allow you to simply resize a partition; you must first delete the partition and then add it back with the desired size. From the FDISK Options screen, choose menu option 3, Delete Partition or Logical DOS Drive, which deletes the necessary partitions. The Delete DOS Partition or Logical DOS Drive screen appears (see Figure 3.3).

Pick the appropriate menu option for the type of partition you're deleting, such as a primary DOS partition. For example, option 1, Delete Primary DOS Partition, allows you to delete primary DOS partitions.

FIG. 3.3
Use the Delete DOS Partition screen to delete a specific partition or logical drive.

Choose option 1 to display the Delete Primary DOS Partition screen (see Figure 3.4). The screen asks for a volume name of the partition and then a confirmation to see whether you

really want to delete the partition. Because all information on the partition will be destroyed, FDISK wants to make absolutely sure that you want to delete the primary DOS partition.

FIG. 3.4
MS-DOS warns you
when you try to delete a
primary DOS partition.


```
                        Delete Primary DOS Partition

Current fixed disk drive: 1

Partition  Status   Type    Volume Label  Mbytes   System   Usage
  C: 1       A    PRI DOS   OPUS_DOS         5     FAT12       4%
     2               Non-DOS                 8                 6%
     3               Non-DOS               376               100%
     4               Non-DOS               114                90%

Total disk space is  127 Mbytes (1 Mbyte = 1048576 bytes)



WARNING! Data in the deleted Primary DOS Partition will be lost.
What primary partition do you want to delete..? [1]
Enter Volume Label...............................? [OPUS_DOS  ]
Are you sure (Y/N)...............................? [N]

Press Esc to return to FDISK Options
```

Adding Partitions   After you delete all the necessary partitions, you must add the appropriate partitions for your DOS system by selecting the Create a DOS Partition menu item on the FDISK Options screen. Figure 3.5 shows the Create a DOS Partition or Logical DOS Drive screen.

FIG. 3.5
Most operating systems
require a primary active
partition to boot
properly.


```
                  Create DOS Partition or Logical DOS Drive

Current fixed disk drive: 1

Choose one of the following:

1. Create Primary DOS Partition
2. Create Extended DOS Partition
3. Create Logical DOS Drive(s) in the Extended DOS Partition



Enter choice: [1]




Press Esc to return to FDISK Options
```

N O T E   You can't add the Linux or OS/2 partitions with the DOS FDISK program. Partitioning the hard drive for Linux is covered later in the section "Using the Linux *fdisk* Command." ▮

Providing all the space available for the partition and making the partition the active partition are the FDISK defaults, as shown in Figure 3.6.

Part

I

Ch

3

FIG. 3.6
You can provide the all
the disk space to one
partition, or spread out
the free space across
several partitions.
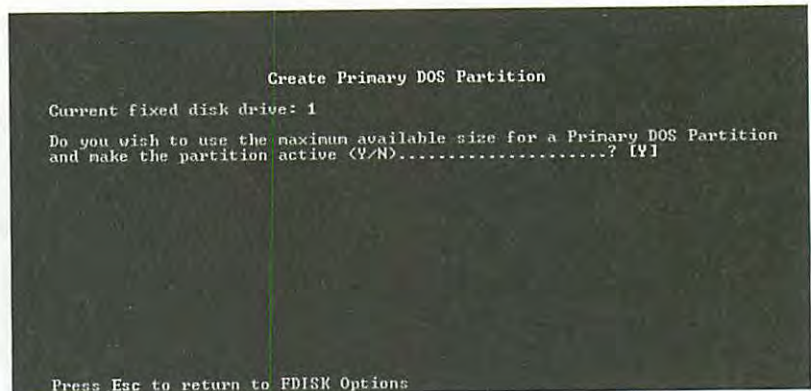

```
                          Create Primary DOS Partition

Current fixed disk drive: 1

Do you wish to use the maximum available size for a Primary DOS Partition
and make the partition active (Y/N)......................? [Y]
```

```
Press Esc to return to FDISK Options
```

Active indicates that the partition is bootable. To boot DOS, you must specify the primary
partition as active. Choose N (no) for this first selection so that you can specify the exact
amount of disk space to provide to your DOS partition. Answering no to the question in Figure
3.6 displays the Specify Disk Space for the Partition screen. Specify the desired space for your
DOS partition either in megabytes or in percentage of space available and press <Return>.

Next, you must set this partition as active. From the FDISK Options screen, choose menu
option 2, Set Active Partition, and simply follow the instructions on the set active menu screen.

Formatting the Partition   After you repartition your hard drive, you need to prepare the new
partition for DOS and restore the appropriate files back to the DOS partition. Reboot your
computer with the boot disk you made earlier. Then format the appropriate drive and transfer
the system files by using the following DOS command:

```
format c: /s
```

When the partition is formatted, you can restore your backup to the new drive. Remember, if
you shrank the partition, not all the files will fit on the new drive. It might be necessary to place
the files that don't fit on the new drive onto other DOS drives or partitions.

Installing the Linux System

To start the Linux installation, place the boot disk you created into your disk drive and reset
your computer. The system displays the following prompt and then begins initializing the
system:

```
LILO:boot:
Loading Linux....
```

Watch the messages as they appear, in case Linux can't recognize your hardware. If Linux
can't, you may have to pass some extra parameters to Linux when it boots up. Record these
parameters, enter them at the prompt, and press <Return>. You should then see a screen ask-
ing if you have a color monitor (see Figure 3.7).

FIG. 3.7

The Red Hat installation program can take advantage of screen colors on a color monitor.
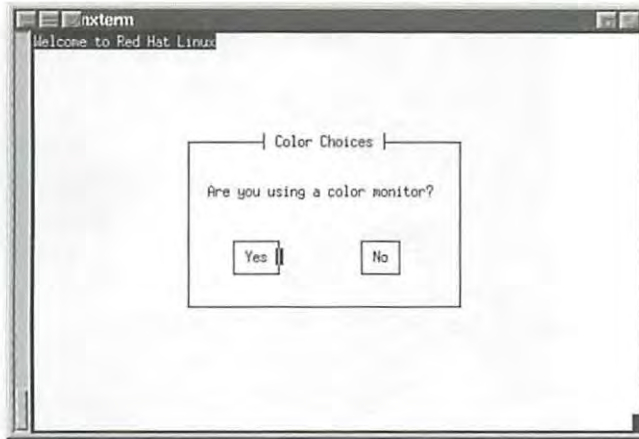
Select your choice by using the <Tab> key and then press <Return> to display the Welcome screen (see Figure 3.8).

FIG. 3.8

The Red Hat Linux Welcome screen.

Press <Return> to continue. The next screen asks whether you need PCMCIA support for your system (see Figure 3.9). Select the appropriate answer with the <Tab> key and press <Return>.

N O T E Moving around in the dialog boxes is easy, and the installation program provides reminders at the bottom of most screens. To move from element to element (field), press <Tab> or <Alt-Tab>. Use the space bar if you need to select an item from a list or check a check box. To choose a button (typically OK or Cancel), just press <Return>.

The next dialog box (see Figure 3.10) asks you to select the installation method described earlier in the section "Understanding the Various Installation Methods." Select your installation method and press <Return>. The installation program asks you to insert the Red Hat CD-ROM into the CD-ROM drive. When you're done, simply press <Return> to continue.

Part
I

Ch
3

FIG. 3.9

Red Hat Linux provides optional support for PCMCIA cards.
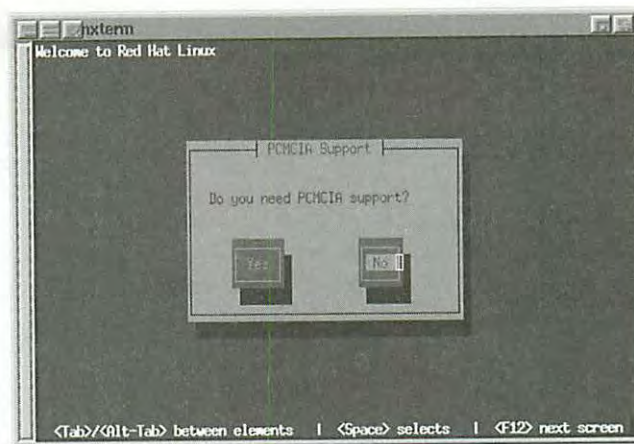
FIG. 3.10

Red Hat lets you install the system by using a variety of methods, including NFS and `ftp`.
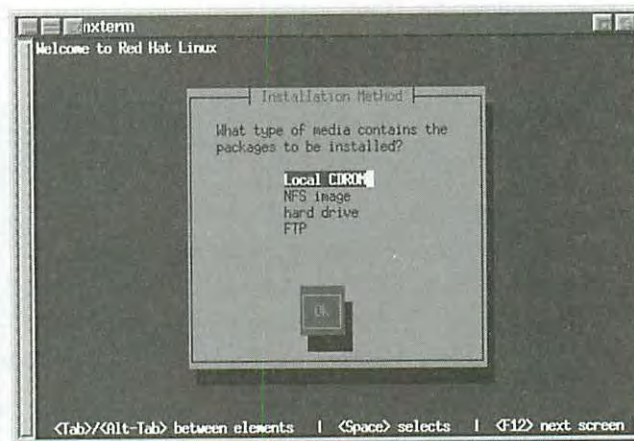
N O T E   The rest of this chapter assumes that you're installing from the local CD-ROM drive. If you select another method of installation, see the appropriate help topics or Red Hat's Web site at **http://www.redhat.com**. ▓

Next, the program asks for your CD-ROM drive type (see Figure 3.11). You can select from

    IDE (ATAPI)

    SCSI

    Other CDROM

FIG. 3.11
The Red Hat Linux
installation program
needs to know what
type of CD-ROM used in
your system.

The Other CDROM category includes such drives as those sold by Creative Labs (SoundBlaster) and other multimedia kit-based CD-ROMs, as well as

Aztech CD	Sanyo
Goldstar R420	Sony CDU-31A
Mitsumi	Sony CDU-5xx
Optics Storage 8000	SoundBlaster/Panasonic
Phillips CM206/CM260	

Depending on your selection, the installation program may ask for some parameters, such as IRQs or DMA addresses. Or the program may try to determined these values automatically by probing your hardware. It's best to let the program autoprobe first before providing parameters.

N O T E Any time the installation program probes the system's hardware, the system may hang. If that happens, you must reboot and redo the installation. Be sure to try to collect the needed information, such as IRQs and DMA addresses, before attempting to reinstall.

After detecting your CD-ROM type, the system starts its installation from the CD-ROM drive. First, it asks you whether you're installing a new system or upgrading an existing Red Hat system. Red Hat 4.0 easily upgrades over version 2.0, 2.1, or 3.0.3, but no Linux distribution easily upgrades over a different distribution version. So if you have a prior distribution version, such as Slackware, just performing a new install and blowing away your prior system is best—after backing up important data files, of course.

Next, the installation program displays the SCSI Configuration dialog box, where you tell the system whether you have any SCSI adapters in your system. Choose the appropriate button and continue.

If you have a SCSI adapter, the program displays the load module dialog box from which you can select from the following SCSI drivers:

Adaptec 152x	Iomega PPA3 (Parallel port Zip)
Adaptec 1542	NCR 5380
Adaptec 1740	NCR 53c406a
Adaptec 2740,2840, 2940	NCR 53C810/53C820 PCI
AdvanSys Adapters	Pro Audio Spectrum/Studio 16
Always IN200	Qlogic FAS
Buslogic Adapters	Qlogic ISP
DTC 3180/3280	Seagate ST01/02
EATA DMA Adapters	Trantor T128/T128F/T228
EATA PIO Adapters)	UltraStor 14F/34F
Future Domain TMC-885, TMC-950	UltraStor 14F/24F/34F
Future Domain TMC-16x0	Western Digital wd7000

Next, you must partition your disks—or at least select the partitions you've already created. You select the drive you want dedicated to Linux in the list box shown in the Partition Disks dialog box, and then press <Return> or choose the Edit button. This drops you into the fdisk program to partition the selected hard drive.

Using the Linux *fdisk* Program

At the fdisk prompt, type m for a list of commands. Table 3.4 lists the available commands.

Table 3.4   The Linux *fdisk* Commands

Command	Description
a	Toggles a bootable flag
c	Toggles the DOS compatibility flag
d	Deletes a partition
l	Lists known partition types
m	Displays this menu
n	Adds a new partition
p	Displays the partition table
q	Quits without saving changes
t	Changes a partition's system ID
u	Changes display/entry units

Command	Description
v	Verifies the partition table
w	Writes the table to disk and exits
x	Provides extra functionality for experts only

To begin the partitioning, select the p command (press <p><Return>) to display the current partition table, which should reflect the drive you partitioned earlier with the DOS FDISK program. Listing 3.1 shows a possible listing from the p command.

Listing 3.1 Example of a Current Partition Table

```
Disk /dev/hda: 15 heads, 17 sectors, 1024 cylinders
Units = cylinders of 255 * 512 bytes

Device        Boot    Begin    Start    End     Blocks    Id    System
/dev/hda1      *       1        1        41      5219      1     DOS 12-bit
FAT
dev/hda2               1024     1024     4040    384667+   51    Novell?
Partition 2 has different physical/logical endings:
phys=(967, 14, 17) Logical=(4096, 14.17)
```

N O T E   Your screen may appear different than what's shown in Listing 3.1, because the values are different for each drive type and the partitions already defined on that drive. ■

Listing 3.1 indicates the various partitions already defined that it can detect, the start and ending locations of the partition, and how big it is in blocks. The listing also indicates the partition type. Table 3.5 shows all the different types of partitions you can define with the Linux fdisk program. The primary partition types you used are 83-Linux Native and 82-Linux Swap. You can get a similar listing with the l command.

Table 3.5 The Known Linux Partition Types

Reference Number	Type	Reference Number	Type
0	Empty	75	PC/IX
1	DOS 12-bit FAT	80	Old MINIX
2	XENIX root	81	MINIX/Linux
3	XENIX usr	82	Linux Swap
4	DOS 16-bit < 32M	83	Linux Native

continues

Table 3.5 Continued

Reference Number	Type	Reference Number	Type
5	Extended	93	Amoeba
6	DOS 16-bit >= 32M	94	Amoeba BBT
7	OS/2 HPFS	a5	BSD/386
8	AIX	b7	BSDI fs
9	AIX bootable	b8	BSDI swap
a	OS/2 Boot Manager	c7	Syrinx
40	Venix 80286	db	CP/M
51	Novell?	e1	DOS access
52	Microport	e3	DOS R/O
63	GNU HURD	f2	DOS secondary
64	Novell	ff	BBT

In Listing 3.1, Linux prints a note about the different physical and logical endings at the bottom of the screen. The difference is because on the system used to write this chapter, a prior partition containing the DOS D drive was left intact, whereas the C drive was repartitioned to a smaller C drive to make room for Linux. Thus, there's space between the C drive and the D drive. This is where the necessary partitions required by Linux will be created.

The begin, start, and end numbers from Listing 3.1 are very important and you should write them down. You'll need them in a later step to specify the necessary sizes of the partitions you'll add.

Adding the Necessary Partition

Because you've repartitioned the drive for DOS, you shouldn't have to delete any partitions for Linux. You only should have to add partitions. To add a partition, issue the n command, which displays

```
Command Action
e extended
p primary(1-4)
```

Press <p><Return>. fdisk then asks for the partition number; enter your selection and press <Return>. If you indicate a partition number already in use, fdisk reports this fact and asks you to delete the partition before trying to add it to the partition table. For this example, enter 3, to add a third primary partition that's referred to as /dev/hda3.

Next, fdisk asks for the location of the first cylinder. This is usually the first available cylinder; in fact, fdisk displays a default range for your selection—for example,

```
First cylinder (42-1024) :
```

Notice that the first partition ends at cylinder 41 and that the next partition begins at cylinder 1024. Thus, the range supplied by fdisk here allows you to start the next partition anywhere in the range of 42-1024. It's a very good idea not to place partitions just anywhere throughout the disk, so choose the next available location, which in this case is cylinder 42. Enter **42** and press <Return>.

N O T E Linux can have trouble booting from partitions defined to start at cylinders above 1024. If you can create a Linux partition only in this range, you may have to boot Linux from a floppy. You learn how to create a boot floppy (different from the boot floppy used for installation) later in this chapter. The only downside is that it takes a little longer to boot Linux from floppy than from the hard drive. ■

Now fdisk wants you to specify how much space to allocate for this partition. You can express this size in number of cylinders or by the number of bytes (+*size*), kilobytes (+*size*K), or megabytes (+*size*M). Because you should already know the approximate size you need for the swap file, define this partition first, and then leave the rest of the disk space for the Linux program partitions. Thus, for this example, your machine has 8M of RAM, so you need to specify a 16M partition size by replying

```
Last cylinder or +size or +sizeM or +sizeK (42-1023): +16M
```

You should then use the p command to look at the new partition table you've defined. In this example, the new partition table looks like

```
Disk /dev/hda: 15 heads, 17 sectors, 1024 cylinders
Units = cylinders of 255 * 512 bytes

Device      Boot Begin Start End   Blocks   Id   System
/dev/hda1    *    1     1     41    5219 1   DOS   12-bit FAT
/dev/hda2         1024  1024  4040  384667+ 51   Novell?

Partition 2 has different physical/logical endings:
phys=(967, 14, 17) Logical=(4039, 14.17)
/dev/hda3         42    42    170   16447+  83     Linux native
```

By default, fdisk made the new partition a Linux Native type. To change this to a swap partition, you need to use the t command. Enter **t** and then enter the partition number you want to change; in this example, enter **3**. fdisk then requests that you enter the hexadecimal value of the desired partition type from Table 3.5 (if you don't have the table handy, you can type **1** and get the list of codes). Because you want a swap partition, enter **82** at the prompt.

As you can see, fdisk reports the new partition type, but you can also use the p command to double-check that partition 3 is now a Linux Swap partition.

Part
I

Ch
3

Now you can add your Linux partitions. For this example, add only one partition, but if you want to have multiple partitions for various reasons, you can also do so at this time. To add a partition, press <n>, specify p for another primary partition, and then specify the number for this partition (4). To keep from fragmenting different partitions across the drive, start the last partition where the other left off, at cylinder 171. For the last cylinder, because you want to use the rest of the space for the Linux system, you can specify the last cylinder instead of an exact byte count. Thus, enter **1023**, as shown here:

```
Command (m for help):n
Command action
e    extended
p    primary partition (1-4)
p
Partition number (1-4): 4
First cylinder (171-1024):171
Last cylinder or +size or +sizeM or +sizeK (171-1023):1023
```

Now use the p command to verify the new partitions. If you need to make any changes, do so now.

When you're satisfied with the layout of your partitions, you can use the w command to write the partition table information to the hard disk. None of your changes are permanent until you use the w command; thus, if you feel you've made some changes in error, you can use the q command to exit without altering the partition table. If you use the w command, Linux tells you the partition table has been altered and then resynchronizes the disks to match the new partition table. If your Linux system hangs at this point, reboot with the installation boot and root disks until you're back at the # prompt.

> **CAUTION**
>
> Don't use the Linux fdisk program to create or modify partitions for other operating systems. This could leave the hard drive in a useless state for both operating systems.

Creating the Swap Partition

Some distributions of Linux provide automatic creation and activation of the swap file during installation, so you don't have to worry about creating the swap file. However, if you're using a different distribution, you may need to create and activate the swap file before continuing with the installation.

N O T E If you get an "out-of-memory" type error during the installation procedures that follow, you should increase the size of your swap file. If you already have the maximum of 16M, you need to create and activate another swap partition by following these instructions. Remember, the Red Hat installation program creates only one swap partition. ▪

To create the swap space, use the mkswap command and tell it which partition to use and how much size to use for virtual RAM. For example, to create a swap space on the /dev/hda3 partition that you previously defined, enter the following command at the # prompt:

```
# mkswap -c /dev/hda3 16447
```

The 16447 represents 16M and can be found in the blocks column of the fdisk p command output screen. The optional -c flag tells mkswap to check for bad sections on the partition.

Next, you need to activate the swap system with the swapon command:

```
# swapon /dev/hda3
```

Again, if you're using the accompanying Red Hat CD-ROM, you shouldn't have to worry about activating the swap system as long as you create the partition for one. During installation, the install program detects the swap partition and automatically starts the system for installation.

After creating your partitions on the various hard drives and returning to the Partitioning Disks dialog box, choose the Done button to continue with the installation.

Next, the system asks you to select the active swap space, which should be the partition you created and marked as type Linux Swap (82) in the preceding section. Select this partition and choose OK. The program then initializes the swap space.

After creating the swap space, the program displays the Select Root Partition dialog box. The root partition is your main file system for Linux where all the boot files are located. Select the device (hardrive) for your root partition from the list box and press <Return>. Now you can mount the other partitions, if any, from the Partition Disk dialog box. From here, you also can mount any DOS or OS/2 file systems so that you can access them from Linux. Select the partition to edit from the list box and press <Return>. From the Edit Mount Point dialog box, you can specify a mount point—that is, a directory—to which you want this partition mounted.

▷ **See** "Mounting and Unmounting File Systems," **p. 265**

After selecting the root and mount points for your various partitions, the program formats those you select. You select the partitions to format from the Format Partition dialog box.

Installing the Software Components

Congratulations! Your system is now prepped for Linux, but you're only halfway finished. Now you must select the various software components to install and then configure them.

The installation program displays the Components to Install dialog box, which allows you to select the various packages. Table 3.6 describes each package.

Table 3.6 The Installation Components

Component	Description
C Development	Provides the GNU gcc compiler and tools
Development Libraries	Provides various libraries needed by the various development tools, such as gcc and g++
C++ Development	Installs the GNU C++ compiler, gcc
Print Server	Allows your Linux box to act as a print server on your network
News Server	Allows your system to act as a news server (if you can get a news feed), thus providing news to your users
NFS Server	Allows your system to export and attach to other file systems on your network
Networked Workstation	Provides networking applications
Anonymous FTP/Gopher Server	Allows you to set up your system so that others may access it via anonymous FTP
Web Server	Includes the most used Web server software today, Apache
Network Management Workstation	Provides utilities and tools to help troubleshoot and monitor your network
Dialup Workstation	Allow you access to the Internet via dial-up lines—that is, via a modem
Game Machine	Provides your system with a variety of text and graphic games, including DOOM
Multimedia Machine	Allows you to play CDs and do other multimedia type actions from your computer
X Windows System	Provides the GUI for all UNIX—hence, Linux—workstations; X is a powerful GUI like Windows 95 and OS/2
X Development	Provides the tools, libraries, and miscellaneous items (such as fonts) needed to develop X applications
X Multimedia Support	Provides multimedia support for X
TeX Document Formatting	Provides a series of programs used to add formatting codes to documents
Emacs	Installs the ubiquitous editor for Linux (you can do anything with emacs, or so the emacs gurus say)
Emacs with X Windows	Provides an X front end to the powerful emacs editor

Component	Description
DOS/Windows Connectivity	Allows you to access DOS files, run DOS programs, and, with limited success, run some Windows programs
Extra Documentation	Provides the Linux documentation project containing the important HOWTOs, along with other helpful information
Everything	Installs everything on the CD-ROM; you need about 350M available, not counting free space for your data files

N O T E You can select individual packages by checking the appropriate check box in the dialog box, or you can install everything by selecting that list option. To select a package to install, simply move to the desired component and press the space bar. After you select all your components, tab to the OK button and press <Return>.

The next dialog box after installation informs you that you can see the files installed by viewing the file /tmp/install.log. Press <Return> to continue with the installation.

Now comes the hard part—waiting. Transferring and decompressing upward of 350M of programs can take a while. Setup firsts installs a file system on your indicated partitions and then starts installing software. The system informs you of its progress as it installs the various files you selected in the Install Status dialog box. Installation time varies depending on what you're installing and how fast your machine can process the information. Relax and order a pizza!

Configuring Your System

After installing the software, the installation program begins to configure your system. It first configures your mouse by displaying the Configure Mouse dialog box. Simply select the mouse type from the list box that best describes your mouse. Remember, many mice can emulate the Microsoft serial mouse if they have to. The Emulate 3 Buttons check box is there because many PC mice have only two buttons, and X Windows usually uses three buttons to maneuver and make program selections. By checking this box, the system will make clicking both mouse buttons at the same time the same as pressing the middle button on a three-button mouse. Make your selection and choose OK.

Next, you must specify the serial port that your mouse connects to. After making the selection from the list box, tab to the OK button and press <Return>. The program then asks you to select the type of video card in your system.

CAUTION

Try to select the correct video card, because of all things software-based, the only subsystem that software can easily destroy is your video card and monitor. If you make the wrong decision, you might fry your monitor! Although this is highly unlikely, there's still the slight possibility, so choose wisely, young Linux walker.

The system now tries to install the proper XFree86 server for your hardware.

▶ **See** "Installing the XFree86 System," **p. 163**

Next, you must select your monitor. Again, be as specific as possible. After selecting your monitor, the program asks for the amount of video memory your card contains. Make the appropriate selection and choose OK to continue.

Remember all those warnings about frying your monitor? Well, now you really have a chance to toast it, so be careful. The next screen asks you to select the clockchips located on your video card. These chips are used to drive the video signals through your card and into your monitor. If they're way out of synchronization, the signals can—you guessed it—fry your monitor (few actually explode, most just fizzle and smoke). Please be careful! If you have no clue as to what clockchips your card is using, take the default selection, No Clockchip Setting, and choose OK.

After selecting your clockchips (or lack thereof), the system can autoprobe and try to configure X. The autoprobe may hang your system, but as long as nothing is seriously wrong (for example, you selected outrageous clock speeds for your card), you can simply reboot and continue with the installation. You do have the option to skip the autoprobe and continue with the installation.

N O T E   I have installed Red Hat many times and have yet to have the installation software properly configure my X system. You may have better luck than me, so don't worry if your X installation fails. I've always been able to configure X with the Xconfig program described in Chapter 6, "Installing the X Windows System." Although the discussion centers on installing under Slackware, installing XFree86 is the same under most Linux distributions, including Red Hat.

▶ **See** "Configuring XFree86," **p. 167** ▪

If the autoprobe succeeds, the system displays an information screen on selecting the resolutions you want to use with your system. You can select more than one, as long as your video card and monitor can handle the resolutions. Finally, the program tells you on how to start and stop your X Windows system.

Configuring Your Network

After configuring X Windows, the installation program continues with your network. If your machine is or will be connected to the Internet and you installed the networking components, choose Yes and continue.

First, the system asks which Ethernet driver to use with the Load Module dialog box. Select the appropriate driver for your Ethernet card and choose OK.

Again, the installation program might try to autoprobe your hardware to determined certain values for the card. This probing can hang the system and force you to reboot. If this happens, hang in there. First make sure that you selected the correct driver. Then see whether you need to pass any special parameters to the device, such as IRQ or DMA address settings. You can do this by selecting the Specify Parameter option instead of the autoprobe option.

N O T E Ethernet is the most popular network interface for Linux today. Other technologies, such as Token Ring, ISDN, and ATM, have some support, but they're not ready for prime time yet under Linux. Many are still in alpha or beta stage and are dependent on vendor-specific hardware.

If the system can detect your network card, it leads you through setting up your TCP/IP network.

Configuring the TCP/IP Network

The installation software uses the Configure TCP/IP dialog box to gather your system's TCP/IP information. Your network administrator or Internet service provider can provide the following information: your machine's IP number, netmask, network address, and broadcast address.

Next, the system must configure your network. It gathers information from the Configure Network dialog box. You must specify your network's domain name and your system's host name. The domain name is typically the last two parts of an Internet address. For example, if the name is www.netwharf.com, then netwharf.com is the domain name and www is the host name.

Next, your network administrator must give you the values for your system's default gateway and the primary name server. Your network may also have a secondary name server, too, so enter the value in the appropriate place.

N O T E Be careful what you name your host, because this name will appear on your default prompt line, in mail messages, and in log reports. Do you really want your boss to receive mail from uradork.netwharf.com?

Configuring the Time

Next, you have to specify how your system will keep time and in what time zone it exists. This is done with the Configure Timezones dialog box. Check whether you want to use local time or GMT time, and then pick your time zone from the list box. After making your selections, choose OK.

Selecting Your Keyboard Settings

After configuring how your system tracks time, the installation program asks what type of keyboard you're using. Use the Configure Keyboard dialog box to select which country code you want to use and choose OK.

Selecting Your Root Password

Now you must select your root password. This is the ultimate key into your system, so some care should be given. The superuser, or root, on a Linux/UNIX system can do great things, and can also wreak awesome damage. Pick a secure password and be careful who you give it to. The Root Password dialog box lets you enter the password twice to confirm what you've

typed. Although you can fix user accounts when they forget their password, if you forget the root password, you may be forced to reinstall the system. However, booting from a floppy and editing the password file may allow a recovery.

▶ **See** "Dealing with Password Security," **p. 239**

Installing LILO

LILO stands for LInux LOader. LILO is a program executed at system startup that lets you choose which operating system is used to boot the computer. You can use LILO to boot several different operating systems, such as Linux and MS-DOS. Press <Tab> to get a list of operating systems LILO can boot.

With LILO, you can specify a default operating system to boot and a default time limit before it boots that system. For example, if you had MS-DOS and Linux on your computer, you can configure LILO to boot either one. You could then tell LILO to boot MS-DOS if no one intervenes after 30 seconds. Before that 30 seconds is up, however, a user can specify another operating system to boot instead of the default. You can press the <Ctrl>, <Alt>, or <Shift> key to stop the timed process.

You specify all this information while configuring LILO. You can later directly edit the lilo.conf file located in the /etc directory. If you don't want to boot Linux automatically, you can select the Skip button to continue. Otherwise, select a hard drive to install LILO and press <Return> to edit the entry.

Congratulations! After you load LILO, your system is up and running—and, let's hope, problem-free.

Building a New Kernel

Sometimes a problem has only one solution—a new kernel. The kernel is the core operating system for Linux. Although not for the faint of heart, downloading a new kernel from the Net and build the kernel is sometimes necessary. If you have some programming experience and know your way around the C programming language, you should be able to build and install a new kernel. If not, you can skip this section.

You may have to install a new kernel for a few reasons:

- A new patch is provided to run new hardware.
- You want to remove features from the kernel you don't use, thus lowering the memory requirements for your system.

The starting point is to determine what kernel version you're now running. You can find out the kernel version with the following command:

```
uname -a
```

The response indicates which version of the kernel is now running and when it was created. The version numbers are in the form of

`MajorVersionNumber.MinorVersionNumber.PatchLevel`

Linus Torvalds is the official release point for new kernels, although anyone can modify Linux (due to the GPL). Linus as the official release point gives the Linux development and user community a common baseline from which to work and communicate.

N O T E   Be sure to read the Kernel HOWTO for up-to-date information before actually trying to build and configure a new kernel.

To build a new kernel, you need to have the source files in the /usr/src/Linux directory. You also need to have the C compiler package loaded, which is disk set d. If you didn't install that package during installation, use RPM to do so now.

First, you must get the new kernel sources or patches. The new sources are usually found on the Internet; check **sunsite.unc.edu** for the latest and greatest kernels. (If you're modifying your current kernel, this step is, of course, unnecessary.) The source files are usually in a tar file and will need to be unarchived.

T I P   Making a backup copy of your current kernel with the following commands is a very good idea:
`cd /usr/src`
`cp Linux linux.sav`
These commands copy the entire Linux source directory to another directory called linux.sav.

Next, you should use the `patch` command to apply any patch files. After preparing the source files, you can configure and build your new system. Start by entering the following command from the /usr/src directory:

`# make config`

The `make` command asks you various questions about the drivers you want to install or configure. Pressing <Return> accepts the default value for each question; otherwise, you must supply the answer. Some of the questions are shown in Table 3.7. You may have to answer other questions depending on the version of the kernel you're installing or the patches you've applied.

Table 3.7   Some Configuration Options

Configuration Option	Description
Kernel Math Emulation	Asks if the kernel should emulate a math coprocessor.
Normal Harddisk Support	Enables drivers for all standard hard drives.

continues

Table 3.7 Continued

Configuration Option	Description
XT Harddisk Support	Applies only if your machine uses an XT class controller instead of an AT class controller.
Networking Support	If you answer yes, enables networking support within the kernel.
SCSI Support	Enables support for SCSI controllers.
CD_ROM Drivers	Asks a series of questions dealing with CD-ROMs, especially those not supported within the standard SCSI Support package.
Filesystems	Asks a series of questions dealing with file systems the kernel should support. If your kernel doesn't have support for the ISO9660 file system, you won't be able to use a CD-ROM.
Parallel Printer Support	Enables printer support via the parallel port.
Mouse Support	Enables support for bus mice from within the kernel.
Sound Card Support	Asks a series of questions about the hardware and software configuration of a sound card.

After you answer the various questions to configure your new kernel, you must compile it. The following commands will build the new kernel:

```
make dep
make clean
make
```

The build process can take anywhere from 15 minutes to many hours. So relax and order another pizza!

When the compilation is completed, you can create a new boot disk as discussed earlier in this chapter. You can copy this kernel to a new floppy disk, or use LILO to boot the new kernel.

Installing Red Hat Linux on DEC Alphas

Unlike other distributions, Red Hat also provides Linux for DEC Alphas. This distribution isn't included on the accompanying Red Hat CD-ROM, but it's available from Red Hat. For more information on getting this distribution, see Red Hat's Web site at **http://www.redhat.com/products/rhl-alpha.html**. When you have the correct distribution, you can use the following instructions to install Red Hat Linux on an Alpha.

N O T E Red Hat also has a distribution for Sun's line of Sun Sparc processors. See Red Hat's Web
site for more information.

Before installing the distribution on an Alpha, you should read the installation instructions in
the first part of this chapter, because many of the steps are the same. You'll also need access to
a computer capable of reading and writing MS-DOS disks, because you must create an installa-
tion floppy.

Using Supported Alpha Hardware

Red Hat supports various Alpha hardware from both Digital Equipment Corporation (DEC)
and other vendors. The following hardware is supported:

- AlphaPC64 (Cabriolet, Aspen Telluride)
- AxpPCI133 (Noname)
- EB64+ (Aspen Alpine)
- EB66 (NekoTek Mach 1)
- EB66+
- Jensen (DEC PC 150, 2000 model 300, Cullean)
- Universal Desktop Box (UDB, aka Multia)
- AlphaStation 200, 250,255,400 (Avanti machines)
- EB164 (Aspen Avalanche, Timerline, Summit)
- Kinetic's Platform 2000 machines
- PC164 (Durango)
- Alcor AlphaStations 500, 600 (Maverick, Brett)
- Alpha-XL
- Alpha-XLT (XL 300, XL 366)
- Mikasa AlphaServer 1000—the 1000A is NOT supported

All of these systems have SCSI systems supported by Red Hat Linux. The video systems
should work too, although S3 support for the Jensen systems isn't included by default. To run
X with a Jensen system, you need to download the X server from **ftp://ftp.azstarnet.com/
pub/linux/axp/jensen**. Finally, all Ethernet solutions for these systems are supported, and
the kernels for these machines also support Token-Ring adapters.

The hardware list changes frequently, so you should check the up-to-date list on Red Hat's Web
site at http://www.redhat.com.

Creating the Boot and Root Disks

You need to create a boot and root floppy for an Alpha installation. The boot disk contains a
program image allowing you to boot the system. The root floppy provides an image of the

Linux kernel for the system to use during installation. Just as for Intel machines, you use the `rawrite` program to create these disk images.

The boot disk image depends on the type of Alpha used. These images are located in the /images directory with a README file that provides more information on each image described in Table 3.8.

Table 3.8   Boot Images Available for DEC Alphas

Image	Description
cab.img	AlphaPC64, Cabriolet
noname.img	AxpPCI33, Noname, Universal Desktop Box (Multias)
eb64p.img	EB64+, Aspen Timberlines
eb66.img	EB66
eb66p.img	EB66+
jensen.img	Jensens
avanti.img	AlphaStation 200, 250, and 400
xl.img	Alpha XL
xlt.img	Alpha XLT
eb164.img	EB164-based machines
p2000.img	Platform 2000
alcor.img	Alcor-based machines
mikasa.img	Mikasa-based machines

To create a boot image for a Universal Desktop Box, you use the command

```
E:\dosutils\rawrite -f E:\images\noname.img -d a: -n
```

where E: represents the drive letter of your CD-ROM. After creating the boot disk, you must create the root disk, which contains the RAM disk image of the Linux kernel. You create the root disk with the following command:

```
E:\dosutils\rawrite -f E:\images\ramdisk.img -d a: -n
```

Installing the Main Red Hat Distribution

After you have your boot media ready, you can install Linux. The installation procedure is very much like the one outlined earlier in the section "Installing the Linux System." The installation program guides you through the process, asking you to make selections from a list of possible choices.

To begin, place your boot disk into the floppy drive and restart your system. At the prompt, enter the following command:

```
boot fd0:vmlinux.gz root=/dev/fd0 load_ramdisk=1
```

You may see several SCSI messages flash by on-screen. Don't worry about them unless you see a message such as scsi0 : 1, which indicates that you have a SCSI termination problem that needs to be fixed before continuing with the installation. If all goes well, you should see the message VFS: Insert Root floppy to be loaded into ramdisk. Insert the root disk you created and press <Return> to continue the installation process.

Going Back to the Beginning

After you complete the setup and configuration of your system, you should reboot the system so that all of your setup and configurations can take effect.

Rebooting Linux is more involved than rebooting DOS—you can't just turn off the power and turn the system back on. If you do so in Linux, you can damage the file structures and systems. Linux tries to repair itself on boot-up. Don't turn off the power while running Linux.

▷ **See** "Performing Backups and Restoring Files," **p. 230**

To exit Linux, use the following command:

```
shutdown [-r] time
```

The optional -r indicates that the system should reboot after shutting down, and time indicates the time that the system should shut down. You can use now in place of time to indicate immediate shutdown. Linux also recognizes the warm boot keys used by DOS to reboot the machine, <Ctrl-Alt-Delete>, which Linux interprets as the command

```
shutdown -r now
```

Make sure that you've removed the root disk from the drive and reboot your new Linux machine.

Troubleshooting Problems

After your machine reboots, the LILO prompt should appear. Make sure that you can boot to your old operating system if you left it on the hard drive. If that system was DOS, press <Shift> and then type the short word you used to identify the DOS partition when you installed LILO. If you enter an invalid word, press <Tab> to get a list of valid operating system types. If you're having problems at this point, place your DOS boot disk in the boot drive and reboot.

You should be able to boot from your boot disk. When your system is up and running under DOS, try the Linux boot disk you created during installation—not the ones you created to originally install the entire system. If that boot disk doesn't work, you may have to reinstall

Linux. Potential problems to check initially are the kernels and your hardware. Before starting over, make sure that you have the appropriate hardware. If you made notes during the installation process, check which kernel you installed against what hardware you have. If you have a SCSI CD-ROM controller, did you install the `idekern` instead of the `scsikern`? But do make sure that you have hardware supported by Linux.

From Here...

When you have your system up and running, you can read the following chapters for further information about Linux:

- Chapter 5, "Running Linux Applications," gets you up to speed on the various programs you just installed.

- Chapter 6, "Installing the X Windows System," provides you with information on installing X if something went amiss during your Red Hat installation. Although designed for the Slackware distribution, the XFree86 software is the same for any Linux distribution version.

- Chapter 7, "Using X Windows," is interesting if you've installed the X system.

- Chapter 13, "Upgrading and Installing Software," provides instructions on how to reinstall packages you may have left out during the initial setup of your Linux system.

- Chapter 26, "Understanding the Internet," briefs you on the basics about how to get to the Internet.

Installing Slackware 96

This chapter gives you the information needed to install the Slackware 96 distribution of Linux. Remember, because Linux isn't a commercial product, you might find some problems. Although this book leads the way, you may find the need to use the resources provided on the CD-ROM. First, you must determine whether you have the appropriate hardware. After you determine that you have the necessary hardware, you must prepare for the installation. Following the steps provided in this chapter should ensure a smooth installation.

This book assumes that you have a working knowledge of DOS and of such items as formatting your hard drive, partition tables, and sector sizes. If this information sounds like a foreign language, check out Que's *Using MS-DOS 6.2*, Special Edition, or ask a computer guru buddy to help you through this. ▨

How to prepare your system for the Installation

Installing Linux requires some preparation, such as formatting your hard drive. You also must plan the layout of your disk space for Linux's optimum performance.

How to partition your hard drive to make room for Linux

Linux requires certain partitions to work effectively, especially if swap space is required. You must also decide whether other operating systems will coexist with Linux and prepare for their installation.

How to Install the typical Slackware distribution

After preparing the system for installation, you must create installation disks, select your installation media (the accompanying Slackware CD-ROM), and begin installation.

What to do If things go wrong

Because Slackware, unlike Red Hat and Caldera, has no formal technical support, you need to know what to do if the installation goes wrong. Remember, unsupported hardware is the most typical reason for installation failure.

> **CAUTION**
> You're about to make major changes to your system, so be careful.

Understanding What You Need to Install Linux

First, you need a distribution of Slackware 96, which is supplied on the accompanying CD-ROM. To start the installation process, you need two formatted high-density floppy disks—either 3 1/2-inch disks or 5 1/4-inch disks.

You should also decide how you intend to boot Linux. You have two choices:

- You can boot Linux from a floppy disk, in which case you need an extra formatted disk—for a total of three disks.

- You can use a program called LILO, the LInux LOader. LILO allows you to specify which operating system to boot. Such programs as OS/2 and Windows NT provide similar functionality. (LILO is discussed later in this chapter.)

Next, you should make sure that you have enough disk space to install Linux. Most people can get by with 200M devoted to Linux—less if you plan not to use such applications as TeX and X Windows.

Having paper and pen nearby is a good idea, so you can take notes just in case something goes wrong; besides, you'll need to jot down some numbers along the way. For configuring XFree86, the X Windows program distributed with Linux, you should write down what type of chipset your video card uses. If you have a serial mouse and modem, write down the serial port that each is using. You need this information later during the configuration process.

Making the Preparations

If you have a brand-new system, or if you have an existing system but don't care what happens to the data already stored on the computer, you can skip most of the following sections and go directly to "Creating the Boot and Root Disks." If, however, you already have a system you're using and simply want to add Linux, you must do some planning because Linux is another operating system, not just a collection of programs.

In general, when you install Linux—a new operating system—you must do the following:

- *Create the Linux boot disks.* The boot disks consist of two floppies that you must create, because you need to bootstrap Linux onto the new system. (The term *bootstrap* comes from the old saying, "Pull yourself up by your bootstraps.")

- *Repartition the hard drive to make room for Linux.* Repartitioning a hard drive may cause problems because it destroys any data contained on the affected partitions.

- *Boot up Linux.* After making room for Linux, you need to boot the Linux system to gain access to the tools required to create its new partitions and file systems.

- *Create the Linux partitions.* Typically, Linux systems need a primary partition to store the files on and a swap file partition, especially if you have a machine with 8M or less of memory.

- *Create the file systems.* A *file system* is basically a section of your hard drive specially formatted to hold files. UNIX and Linux use file systems to represent entire sections of the directory tree. This is in contrast to MS-DOS, which places subdirectories in the directory tree on the same logical drive. UNIX systems use the file-system structure because placing subdirectories on different drives is safer. If one drive malfunctions, only the information on that drive needs to be replaced or fixed.

- *Install the Linux system and software applications.* After creating the file systems, you install the Linux operating system, its support files, and various application packages distributed with the system, such as the games and networking support packages.

Preparing Floppies and Backups for Installation

You must make a system disk for your PC. To install Linux, you need to repartition your hard drive to make room for the new operating system. Unfortunately, you just can't simply copy the files over to an MS-DOS, OS/2, or Windows NT file system.

Creating an MS-DOS Boot Disk

When you repartition your hard drive, you should prepare for the worst—the trashing of your hard drive. If your drive crashes, you may not be able to boot up. Thus, you should create a boot floppy with the necessary boot files.

For MS-DOS, place a floppy into the disk drive and type **format a: /s** to create an MS-DOS boot floppy. Copy your current CONFIG.SYS, AUTOEXEC.BAT, and any driver files needed to start your computer. Also, copy FDISK.EXE and FORMAT.COM onto the floppy, because you'll be repartitioning your hard drive and will then need to reformat the hard disk. Keep this MS-DOS boot disk in a safe place.

N O T E If you're backing up your system (two backups are a good idea), also include the necessary file to restore your backup.

You might want to record the various CMOS settings for your system. Most Intel-compatible computer systems allow you to access the setup information from the CMOS at startup.

N O T E If you're using OS/2, you need to see the OS/2 documentation on how to create an OS/2 boot disk.

Backing Up Your System

Unless you're using a brand-new computer or a computer with a brand-new hard drive containing no software, take the time to make a backup; two backups is the wisest course of action. If

you have a tape drive, use the program sold with your tape drive to back up the entire hard drive. If you don't have a tape drive, you can use the MS-DOS backup/restore programs to make a backup of your system. Make sure that you have plenty of floppy disks and use the following command syntax:

`backup` *from to*

For example, if you're going to back up all the files *from* the C drive *to* the A drive, enter **backup c: a:** and start swapping floppies. Of course, with the cost of large hard drives dropping, as well as the cost of tape backup units, you should consider doing a tape backup instead of backing up to floppies.

Creating the Boot and Root Disks

After you make your DOS boot disk and back up your system, you must create the boot disks for Linux. Two floppies are needed with Slackware: the boot and root floppies. You create these floppies with a set of MS-DOS programs provided with most Linux distributions—`gzip` and `rawrite`.

> **N O T E** The boot images are decompressed on most major Linux CD-ROMs, including the Slackware CD-ROM that comes with this book. Also, the root images have new names compared with earlier versions of Slackware and are now compressed and *must remain compressed*; otherwise, you'll get a kernel panic because you shouldn't have decompressed the root image files. ■

`gzip` is a GNU program very much like PKWare's `pkzip`. `gzip` allows you to compress and decompress archived files. You need this program because many of the Linux files—especially those needed for installation—are compressed by using `gzip`.

`rawrite` is a program used to write the contents of a file directly to a floppy without regard to the format. After you unzip the boot images with `gzip`, you must use `rawrite` to transfer the images to the appropriate floppies.

Fortunately, most of the common boot and root disks can be found on the Slackware CD-ROM in decompressed format. All you have to do is use `rawrite` to transfer the files to floppy.

> **N O T E** If you're having trouble creating your boot and root disks or getting Linux to boot, read the readme.txt and which.one files in the appropriate boot disk directory (/bootdsks.12 or /bootdsks.144) found on the Slackware CD-ROM. ■

You use the boot disk to start the Linux system for installation. It contains bare-bones device drivers and a basic version of the operating system. This boot disk is specific to the hardware and type of floppy drive your system uses to boot (usually the A drive under MS-DOS). Table 4.1 shows the various files. (These files may have been updated before the printing of this book. See the readme file in the /bootdsks.12 or /bootdsks.144 directory for details.) If the files are already decompressed (if they don't have the .gz extension), you don't have to use `gzip` to prepare that file.

For Slackware 96, also known as Slackware Linux 3.1.0, the Linux 2.0.0 series of kernels is used. All you have to do to create your boot disk is first select your kernel based on Table 4.1 for IDE drives or Table 4.2 for SCSI drives. Then issue the following command from the DOS prompt:

```
rawrite bares.i a:
```

Table 4.1 The IDE Boot Files

Name	Descriptions
aztech.i	CD-ROM drives: Aztech CDA268-01A, Orchid CD 3110, Okano/ Wearnes CDD110, Conrad TXC, CyCDROM CR520, CR540
bare.i	None, just IDE support
cdu31a.i	Sony CDU31/33a CD-ROM
cdu535.i	Sony CDU531/535 CD-ROM
cm206.i	Philips/LMS cm206 CD-ROM with cm260 adapter card
goldstar.i	Goldstar R420 CD-ROM (sometimes sold in a Reveal Multimedia Kit)
mcd.i	NON-IDE Mitsumi CD-ROM support
mcdx.i	Improved NON-IDE Mitsumi CD-ROM support
net.i	Ethernet support
optics.i	Optics Storage 8000 AT CD-ROM (the "DOLPHIN" drive)
sanyo.i	Sanyo CDR-H94A CD-ROM support
sbpcd.i	Matsushita, Kotobuki, Panasonic, CreativeLabs (SoundBlaster), Longshine, and Teac non-IDE CD-ROM support
xt.i	MFM hard drive support

Table 4.2 The SCSI Boot Files

Name	Descriptions
7000fast.s	Western Digital 7000FASST SCSI support
advansys.s	AdvanSys SCSI support
aha152x.s	Adaptec 152x SCSI support
aha1542.s	Adaptec 1542 SCSI support
aha1740.s	Adaptec 1740 SCSI support

continues

Table 4.2 Continued

Name	Descriptions
aha2x4x.s	Adaptec AIC7xxx SCSI support for these cards: AHA-274x, AHA-2842, AHA-2940, AHA-2940W, AHA-2940U, AHA-2940UW, AHA-2944D, AHA-2944WD, AHA-3940, AHA-3940W, AHA-3985, AHA-3985W
am53c974.s	AMD AM53/79C974 SCSI support
aztech.s	All supported SCSI controllers, plus CD-ROM support for Aztech CDA268-01A, Orchid CD-3110, Okano/Wearnes CDD110, Conrad TXC, CyCDROM CR520, CR540
buslogic.s	Buslogic MultiMaster SCSI support
cdu31a.s	All supported SCSI controllers, plus CD-ROM support for Sony CDU31/33a
cdu535.s	All supported SCSI controllers, plus CD-ROM support for Sony CDU531/535
cm206.s	All supported SCSI controllers, plus Philips/LMS cm206 CD-ROM with cm260 adapter card
dtc3280.s	DTC (Data Technology Corp) 3180/3280 SCSI support
eata_dma.s	DPT EATA-DMA SCSI support (boards such as PM2011, PM2021, PM2041, PM3021, PM2012B, PM2022, PM2122, PM2322, PM2042, PM3122, PM3222, PM3332, PM2024, PM2124, PM2044, PM2144, PM3224, PM3334)
eata_isa.s	DPT EATA-ISA/EISA SCSI support (boards such as PM2011B/9X, PM2021A/9X, PM2012A, PM2012B, PM2022A/9X, PM2122A/9X, PM2322A/9X)
eata_pio.s	DPT EATA-PIO SCSI support (PM2001 and PM2012A)
fdomain.s	Future Domain TMC-16x0 SCSI support
goldstar.s	All supported SCSI controllers, plus Goldstar R420 CD-ROM (sometimes sold in a Reveal Multimedia Kit)
in2000.s	Always IN2000 SCSI support
iomega.s	IOMEGA PPA3 parallel port SCSI support; also supports the parallel port version of the ZIP drive
mcd.s	All supported SCSI controllers, plus standard non-IDE Mitsumi CD-ROM support
mcdx.s	All supported SCSI controllers, plus enhanced non-IDE Mitsumi CD-ROM support
n53c406a.s	NCR 53c406a SCSI support
n_5380.s	NCR 5380 and 53c400 SCSI support

Name	Descriptions
N_53c7xx.s	NCR 53c7xx, 53c8xx SCSI support (most NCR PCI SCSI controllers use this driver)
optics.s	All supported SCSI controllers, plus support for the Optics Storage 8000 AT CD-ROM (the "DOLPHIN" drive)
pas16.s	Pro Audio Spectrum/Studio 16 SCSI support
qlog_fas.s	ISA/VLB/PCMCIA Qlogic FastSCSI! support (also supports the Control Concepts SCSI cards based on the Qlogic FASXXX chip)
qlog_isp.s	Supports all Qlogic PCI SCSI controllers, except the PCI-basic, which is supported by the AMD SCSI driver
sanyo.s	All supported SCSI controllers, plus Sanyo CDR-H94A CD-ROM support
sbpcd.s	All supported SCSI controllers, plus Matsushita, Kotobuki, Panasonic, CreativeLabs (SoundBlaster), Longshine, and Teac non-IDE CD-ROM support
scsi.s	A generic SCSI boot disk, with support for most SCSI controllers that work under Linux
scsinet.s	All supported SCSI controllers, plus full Ethernet support
seagate.s	Seagate ST01/ST02, Future Domain TMC 885/950 SCSI support
trantor.s	Trantor T128/T128F/T228 SCSI support
ultrastr.s	UltraStor 14F, 24F, and 34F SCSI support
ustor14f.s	UltraStor 14F and 34F SCSI support

These boot files come in two flavors, depending on the size of your boot floppy. Luckily, the decompressed files are already available for you on the Slackware CD-ROM. If your A drive is a 1.2M (high-density) 5 1/4-inch drive, look in the /bootdsks.12 directory; if you have a 1.44M (high-density) 3 1/2-inch high-density drive, look in the /bootdsks.144 directory. If you use another distribution version, you need to copy the compressed files to a directory on your hard drive and use the `gzip` program to unzip them. Table 4.3 illustrates the commands for the `gzip` program, which uses the syntax

```
gzip [-acdfhlLnNtvV19] [-S suffix] [file ...]
```

Table 4.3 Flags for the *gzip* Command

Flag	Flag Name	Description
-a	ascii	ASCII text; converts end-of-line characters by using local conventions
-c	stdout	Writes on standard output, keeps original files unchanged

continues

Table 4.3   Continued

Flag	Flag Name	Description
-d	decompress	Decompresses
-f	force	Forces overwrite of output file and compresses links
-h	help	Gives a help listing
-l	list	Lists compressed file contents
-L	license	Displays software license
-n	no-name	Doesn't save or restore the original name and time stamp
-N	name	Saves or restores the original name and time stamp
-q	quiet	Suppresses all warnings
-S suffix	suffix .suf	Uses suffix .suf on compressed files
-t	test	Tests compressed file integrity
-v	verbose	Changes to verbose mode
-V	version	Displays the version number
-1	fast	Compresses faster
-9	best	Compresses better—that is, the file is smaller
file		Specifies file(s) to (de)compress; if none given, uses standard input

To create the images for a SoundBlaster installation, you unzip the sbpcd.gz file with the following command:

```
gzip -d sbpcd.gz
```

The result is a file called sbpcd, without the .gz extension. As stated earlier, the accompanying CD-ROM already contains the decompressed files in the appropriate boot directory.

Making the Root Disks   The Linux "root disk" contains the actual installation software. You use a boot disk to load it into your system and begin the installation. You'll probably want to use either the color or the UMSDOS disk image to install Linux, depending on whether you'll be installing it in its own partition. You need one of the image files listed in Table 4.4 for your system.

Table 4.4   Slackware 96 Root Images

File	Description
color.gz	The disk you'll want to use to install Linux onto its own partition.

Petitioners Microsoft Corporation and HP Inc. - Ex. 1012, p. 80

File	Description
umsdos.gz	The disk used to install Linux into an existing MS-DOS partition (for example, into a C:\LINUX directory).
text.gz	A text-based disk that installs Linux into its own partition. This disk's text interface might be a little harder to use than the color disk, but better diagnostics are printed, and you can scroll back to look at messages with <Shift-Page Up>.
pcmcia.gz	A text-based root disk used to install Linux via PCMCIA Ethernet.
rescue.gz	A disk with an editor (vi), LILO, e2fsck, and other utilities useful for repairing a misconfigured Linux system. If you ever get locked out of your system for some reason, you can use this disk to mount your Linux partitions and fix the problems.

Creating the Disks with *rawrite*   After you decompress the kernel image file, you can transfer the file to floppies. To do this, you need the rawrite program. You can find this program on the accompanying Slackware CD-ROM in the /install subdirectory.

For this step, you need the two floppies you formatted earlier. You must make sure that you write the current images to the floppy disks. Thus, you must write the images made for 1.44M disks to 1.44M disks. If you write the images from the 1.2M directories to a 1.44M disks, you won't be able to install Linux.

Label one disk the boot disk and place it in the disk drive. The following sequence assumes a 1.44M, 3 1/2-inch floppy disk drive, sbpcd.i(boot), and color144(root) installation disks (these commands create the boot and root disks):

```
E:\LINUX>RAWRITE sbpcd.i
Number of sectors per track for this disk is 18
Writing image to drive A:,  Press ^C to abort.
Track: 01  Head:  1 Sector: 10
```

rawrite displays progress information as the program writes the raw image to the disk, and then the program displays done when finished. If you want to abort the process, simply press <Ctrl-c>. If any errors occur during the operation, the program displays one of the error messages shown in Table 4.5.

Table 4.5 Possible *rawrite* Error Codes

Message	Description
Operation Successful	Not an error message; you want to see this message!
Bad command	Your hardware I/O system didn't understand a command sent by the program. Try a different version of rawrite from another distribution package.
Address mark not found	Try a different disk.

continues

Table 4.5   Continued

Message	Description
Attempt to write on the write-protected disk	Remove the write protection from the disk.
Sector not found	Possibly a bad disk; try a different one.
Reset failed (hard disk)	Possible hardware problems; check hardware before continuing.
Disk changed since last operation	Don't remove the disk from the drive until the entire operation is complete.
Drive parameter activity failed	Possible hardware problems.
DMA overrun	Data was lost on transfer from memory to disk; try the operation again. This may indicate hardware or low memory problems.
Attempt to DMA across 64K boundary	Attempt by DMA chip to write beyond a 64K boundary. May indicate hardware or low memory problems.
Bad sector detected	Disk is bad; try a new one.
Bad track detected	Disk is bad; try a new one
Unsupported track	Possibly a bad disk; try a new one.
Bad CRC/ECC on disk read	Bad disk; try a new one.
CRC/ECC corrected data error	Possibly a bad disk or hardware.
Controller has failed	Major hardware problems.
Seek operation failed	Possibly a bad disk or hardware problem.
Attachment failed to respond	Check drive door; make sure that it's closed.
Drive not ready (hard disk only)	Possible major hardware problems.
Undefined error occurred (hard disk only)	Possible hardware problems.
Write fault occurred	Possibly a bad disk or hardware problems.
Status error	Undefined error—could be hardware or disk.
Sense operation failed	Possible bad disk or hardware problem.

If rawrite fails, try a new formatted disk. If the problem persists, you should have your hardware checked for possible problems. After writing the boot disk, you need to write the root

disk to your other formatted disk. Simply use the root image file name as the source file name instead of the boot image file name:

```
E:\LINUX>RAWRITE color.gz
Number of sectors per track for this disk is 18
Writing image to drive A:, Press ^C to abort.
Track: 01  Head:  1 Sector: 10
```

Partitioning Your Hard Drive

After you back up your system and make the necessary disks, you must prepare your system's hard drive for Linux.

CAUTION

This process is the most dangerous because maximum data loss is assured. If you haven't backed up your system, do so now. Although there's an experimental program called FIPS and commercial programs such as Partition Magic that do non-destructive repartitioning, a full backup is recommended (two backups are best), just in case problems occur.

Understanding Partitions

In the early days of PCs, hard drives were few and far between. Most computers used floppies to hold the operating system, programs, and their data. With the introduction of the IBM PC XT, IBM introduced a 10M hard drive. Early operating systems such as DOS could access only a limited amount of space on hard drives. Then hard-drive manufacturers kept expanding the space on their hard drives more quickly than the operating system's capability to access the additional space. The operating system got around this problem by letting the user split the hard drive into sections, called *partitions*. These partitions can hold program files, other operating systems, or data.

Typical MS-DOS systems have one partition, which is referred to as drive C. If you split the drive into partitions, these partitions are typically referred to in alphabetical order as D, E, and so on. MS-DOS also allows you to install multiple hard drives, so that the next drive in this chain might be referred to as F.

UNIX and Linux don't use drive letters to refer to partitions; instead, they use directory names to refer to partitions. Also, as indicated earlier, Linux users can place different directories on different partitions and even on different drives. You can also place different operating systems on different partitions.

Partitions are specified in a section of the hard drive referred to as the *boot record* in what's called a *partition table*. This table is used by the various operating systems to determine which operating system to boot and where each OS's files can physically be found on the hard drive. The boot record is used to boot, or start up, the machine's operating system. LILO and other boot managers use this section of the hard drive—typically found on the first sectors of the drive—to control which operating system to start.

The partition table holds information about the locations and sizes of the various partitions on the hard drive. There are three kinds of partitions: primary, extended, and logical. DOS and some other operating systems must boot from primary partitions. Hard drives can contain only four primary partitions. An extended partition doesn't contain data itself; instead, it allows the user to define other, logical partitions on the drive. Thus, to get around the limit of four primary partitions, you can define an extended partition and then define other logical partitions within the extended partition. Some operating systems, such as MS-DOS and versions of OS/2 before version 2.0, require that they be installed in a primary partition, but they can access logical drives in extended partitions. This is important to remember if you're going to have a DOS system and a Linux system reside on the same drive. DOS must go in a primary partition.

Using *FDISK*

Partitions are created, destroyed, and managed by a program usually called FDISK. Each operating system has its own version of FDISK, so be sure to use the correct one. If you're now using MS-DOS or are planning to use MS-DOS, you must first repartition the DOS drive with DOS's FDISK. You later use the Linux version of fdisk to create the Linux partitions. If you're using OS/2, you also need to use OS/2's version of FDISK to prepare the OS/2 partitions.

> **CAUTION**
>
> Don't use the Linux fdisk program to create or modify partitions for other operating systems. This could leave the hard drive in a useless state for both operating systems.

Partition Requirements   First, you should plan out what partitions you need. MS-DOS requires a primary partition; Linux and OS/2 can reside in other partitions. If you're using the OS/2 boot manager, which also works well with Linux, you must prepare for its use also. You must also be aware that if you're shrinking a current MS-DOS partition to make room for Linux, not all of your files can be restored to the new, smaller MS-DOS partition.

You can access MS-DOS partitions from Linux, and move, save, and edit DOS files under Linux. However, you can't execute DOS programs under Linux.

N O T E   Two experimental components of Linux allow you to emulate DOS under Linux and also install Linux under DOS. Both systems are still in the implementation stage and thus are more suited for Linux hackers. You can also find plenty of information on these topics in the Linux world.

Next, you should jot down the number of partitions you need and how much disk space to provide each with.

DOS Requirements   If you want to boot DOS, it must go in a primary partition. A bootable version of DOS doesn't require much space—just enough for the system files, COMMAND.COM, CONFIG.SYS, and any driver files needed to start your system. For

instance, I provide a 5M DOS partition on my first drive to boot DOS. As soon as DOS is loaded and running, you can access any of the other extended and logical drives on the system.

Unfortunately, whereas Linux can access DOS files in a DOS partition, DOS can't access Linux files in a Linux partition.

OS/2 Requirements   OS/2 versions 2.0 and later don't need a primary partition; these versions can install and boot from an extended partition. Thus, you can install DOS on a primary partition and create an extended partition area for OS/2 and Linux.

The space required for OS/2 is version- and feature-dependent; you should consult your OS/2 documentation for space requirements. You should also subtract 1M from available space if you intend to use the OS/2 boot manager.

Linux Requirements   As explained earlier, Linux stores files on file systems, and these file systems can reside on different partitions, basically as safety precautions. Linux requires one partition for each file system. The next consideration is for a swap partition. Linux, like most operating systems that use disk space for memory (called a *virtual memory configuration*), needs a swap file or a swap partition to simulate physical memory using disk space. Linux typically uses a swap partition.

The size of the swap partition depends on the amount of physical RAM your system contains. A rule of thumb is to make your swap partition twice the size of your amount of RAM. Thus, if you have 8M of RAM in your system, you should create a swap partition 16M in size. If you have 4M of RAM or less, you must activate a swap partition.

Linux swap partitions can be only 128M in size, so if you need more space, you have to create multiple swap partitions. For example, if you have a Linux system with 32M of RAM and want one partition for Linux system files and one partition for user files, you would create three partitions: one for the system files, one for user files, and one 32M swap partition.

Repartitioning the Drive

This section assumes that you need to repartition a DOS drive. First, you execute FDISK by typing **fdisk** at the DOS prompt, and you see the FDISK Options screen (see Figure 4.1).

FIG. 4.1
From the FDISK Options screen, you can look at current partitions, create new partitions, and delete old partitions.


```
                          MS-DOS Version 6
                       Fixed Disk Setup Program
                  (C)Copyright Microsoft Corp. 1983 - 1993

                          FDISK Options

Current fixed disk drive: 1

Choose one of the following:

1. Create DOS partition or Logical DOS Drive
2. Set active partition
3. Delete partition or Logical DOS Drive
4. Display partition information
5. Change current fixed disk drive

Enter choice: [1]




Press Esc to exit FDISK
```

The screen shown in Figure 4.1 might appear to look different, depending on which version of MS-DOS you're using. Pick menu option 4, Display Partition Information. The Display Partition Information screen appears (see Figure 4.2).

FIG. 4.2
The Display Partition Information screen in MS-DOS 6.x shows the current partition information.

Write down the information on this screen. You need the current partition table information if you decide to abort the Linux installation and put your system back the way it was before you started.

Alternatives to Repartitioning Your Hard Drive

You may not need to repartition your hard drive, although it's thought that repartitioning offers the best introduction to Linux. You can either use FIPS to repartition your hard drive non-destructively, or install Linux on the same partition as DOS with UMSDOS.

FIPS stands for *First non-destructive Interactive Partition Splitting*, a program developed by Amo Schaefer as a result of the Linux project. It's used to move DOS partitions around to make room for Linux partitions. All instructions for using FIPS can be found in the fips.doc document in the /install/ fips directory on the accompanying Slackware CD-ROM. FIPS can help only if you have enough free space left on your hard drive to install Linux; otherwise, you either need to delete unneeded files or use the process described earlier to repartition your hard drive.

UMSDOS is a project to allow Linux to exist on DOS partitions. UMSDOS allows you to create the Linux root file system under an existing DOS directory. You learn more about using UMSDOS later in the section "Using UMSDOS Instead of Formatting Your Hard Drive."

Deleting Partitions   Unfortunately, FDISK doesn't let you simply resize a partition—you must first delete the partition and then add it back with the desired size. From the FDISK Options screen, choose menu option 3, Delete Partition or Logical DOS Drive, to delete the necessary partitions. If you press <Esc> at anytime, you are returned to the previous screen. The Delete DOS Partition or Logical DOS Drive screen appears (see Figure 4.3).

Pick the appropriate menu option for the type of partition you're deleting, such as a primary DOS partition. If, for example, you choose option 1 (Delete Primary DOS Partition), you see the Delete Primary DOS Partition screen (see Figure 4.4).

FIG. 4.3
Use the Delete DOS
Partition screen to
delete a specific
partition or logical drive.

```
                    Delete DOS Partition or Logical DOS Drive

Current fixed disk drive: 1

Choose one of the following:

1.   Delete Primary DOS Partition
2.   Delete Extended DOS Partition
3.   Delete Logical DOS Drive(s) in the Extended DOS Partition
4.   Delete Non-DOS Partition


Enter choice: [1]




Press Esc to return to FDISK Options
```

FIG. 4.4
MS-DOS warns you
when you try to delete a
primary DOS partition.

```
                      Delete Primary DOS Partition

Current fixed disk drive: 1

Partition  Status    Type    Volume Label  Mbytes   System    Usage
  C: 1       A      PRI DOS    OPUS_DOS        5      FAT12      4%
     2               Non-DOS                   8                 6%
     3               Non-DOS                 376               100%
     4               Non-DOS                 114                90%

Total disk space is  127 Mbytes (1 Mbyte = 1048576 bytes)



WARNING! Data in the deleted Primary DOS Partition will be lost.
What primary partition do you want to delete..? [1]
Enter Volume Label.............................? [OPUS_DOS  ]
Are you sure (Y/N).............................? [N]

Press Esc to return to FDISK Options
```

The screen asks for a volume name of the partition and then a confirmation to see whether you
really want to delete the partition. Because all information on the partition will be destroyed,
FDISK wants to be absolutely sure that you want to delete the primary DOS partition.

Adding Partitions After you delete all the necessary partitions, you must then add the appro-
priate partitions for your DOS system. You can't add the Linux or OS/2 partitions with the DOS
FDISK program (partitioning the hard drive for Linux is covered later in this chapter). Figure
4.5 shows the Create a DOS Partition or Logical DOS Drive screen.

Providing all the space available for the partition and making the partition the active partition
are the FDISK defaults (see Figure 4.6). *Active* indicates that the partition is bootable. To boot
DOS, you must specify the primary partition as active. Say no to this first question so that you
can specify the exact amount of disk space to provide to your DOS partition. Answering No to
the question in Figure 4.6 displays the Specify Disk Space for the Partition screen. Specify the
desired space for your DOS partition in megabytes or in percentage of space available and
press <Return>.

Part

I

Ch

4

FIG. 4.5

To boot properly, most operating systems require a primary active partition.

```
                    Create DOS Partition or Logical DOS Drive
Current fixed disk drive: 1

Choose one of the following:

1. Create Primary DOS Partition
2. Create Extended DOS Partition
3. Create Logical DOS Drive(s) in the Extended DOS Partition


Enter choice: [1]




Press Esc to return to FDISK Options
```

FIG. 4.6

You can allot all the disk space to one partition, or spread out the free space across several partitions.

```
                    Create Primary DOS Partition
Current fixed disk drive: 1

Do you wish to use the maximum available size for a Primary DOS Partition
and make the partition active (Y/N).....................? [Y]











Press Esc to return to FDISK Options
```

Next, you must set this partition active. From the FDISK Options screen, choose menu option 2, Set Active Partition, and simply follow the instructions on the Set Active menu screen.

Formatting the Partition   After you repartition your hard drive, you need to prepare the new partition for DOS and restore the appropriate files back to the DOS partition. Reboot your computer by using the boot disk you made earlier. Then format the appropriate drive and transfer the system files by using the following command:

```
format c: /s
```

When the partition is formatted, you can restore your backup to the new drive. Remember, if you shrank the partition, not all the files will fit on the new drive. It might be necessary to place the files that don't fit on the new drive onto other DOS drives or partitions.

Using UMSDOS Instead of Formatting Your Hard Drive

UMSDOS is a full-featured UNIX file system Slackware Linux can use that resides on an MS-DOS partition, meaning that you don't have to repartition your hard drive. The name stands for *UNIX* in *MS-DOS*. UMSDOS acts within the constraints of the MS-DOS file system while still

providing the long file names as well as other file system attributes known to the UNIX world. However, you can't use both your MS-DOS areas on the hard drive and your UMSDOS files at the same time. Of course, by using various Linux utilities and taking advantage of the advanced file system, Linux can access your MS-DOS areas. Under MS-DOS, you'll be able to see the Linux files, but some may not make much sense, because MS-DOS can recognize only 12-character file names (the eight-dot-three scheme) and can't handle multiple periods in a file name, like Linux can do.

> **CAUTION**
>
> When you use MS-DOS, you can see a file named `--linux-.---`. *Do not erase this file!* This file maintains information used by Linux to handle such items as long file names and symbolic links. Deleting this file from MS-DOS will severely damage your Linux file system!
>
> Windows 95 also allows long file names, but at this moment, Windows 95 and UMSDOS have problems working together. If you try to install UMSDOS under Windows 95, beware—you could trash your entire system!

To install the UMSDOS system, first create a Linux root disk containing the UMSDOS kernel. This is the umsdos.gz file, which works with 1.44M or 1.2M floppies. Follow the steps detailed earlier in the section "Making the Root Disks." Next, specify the MS-DOS partition table or an empty directory on an existing MS-DOS partition on which to install the UMSDOS file system.

▶ See "Understanding File Systems," **p. 262**

▶ See "File Types," **p. 289**

Part

I

Ch

4

Preparing the Hard Drive for Linux

After you repartition your hard drive and make the Linux boot and root disks, or after you prepare a UMSDOS file system, you need to install Linux. You must first boot Linux from the disks you've just created. Then you need to use the Linux version of fdisk to create the necessary Linux partitions. After preparing the hard drive for installation, you can run the Setup program to install and configure Linux to your taste.

Booting Linux

To use the boot and root disks you made, simply place the boot disk into the drive and reboot your system. The following screen appears:

```
Welcome to the Slackware 96 Linux (v.3.1.0) bootkernel disk!

If you have any extra parameters to pass to the kernel, enter them
at the prompt below. For instance, you might need something like
this to detect the hard drives on PS/1 and ValuePoint models from IBM:

ramdisk hd=cyl,hds,hds,secs
  (where "cyl", "hds", and "secs" are the number of
cylinders, sectors, and heads on the drive.)
```

```
Also, in a pinch, you can boot your system with a command like:
mount root=/dev/hda1

On machines with low memory, you can use mount root=/dev/fd1 or
mount root=/dev/df0 to install without a ramdisk. See LOWMEN.TXT for Details.

If you would rather load the root/install disk from your second
floppy drive:
drive2

DON'T SWITCH ANY DISKS YET! This prompt is just for entering
extra parameters. If you don't need to enter any parameters,
hit ENTER to continue.

boot:
```

If you have any parameters to pass to the kernel before it boots, enter them at the boot:
prompt. (For information on these parameters, see the BootPrompt HOWTO in /doc/
HOWTO.) If you don't have any parameters, press <Return> to continue the installation.

Linux then loads itself into a RAM disk and decompresses before running. While booting,
Linux displays information to let you know what's happening, including what type of hardware
it can detect and use. The system then checks the amount of memory available, which floppy
drives are available, and what network protocols are available. Next, Linux checks the floating-
point processor, if available, and how the two processors communicate. Linux then displays the
identity, date, and time at which the Linux kernel was built. Then Linux does a partition check
on the available partitions. Finally, because Linux was booted from the boot disk, it displays the
following:

```
VFS: Insert root floppy disk to be loaded into ramdisk and press ENTER.
```

Replace the boot disk with the root disk and press <Return>.

The system starts by displaying information messages from the RAM disk, indicating how
many bytes of information it's installing and what address it's using. When finished, the system
displays the following information:

```
Welcome to the Slackware '96 Linux installation disk!, (v. 3.1.0)

###### IMPORTANT! READ THE INFORMATION BELOW CAREFULLY. ######
- You will need one or more partitions of type "Linux Native" prepared.
It is also recommended that you create a swap partition (type
"Linux Swap") prior to installation. Most users can use the Linux
"fdisk" utility to create and tag the types of all these partitions.
OS/2 Boot Manager users, however, should create their Linux partitions
with OS/2 "fdisk", add the bootable (root) partition to the Boot Manager
menu, and then use the Linux "fdisk" to tag the partitions as type
"Linux native".
- If you have 4 megabytes or less of RAM, you MUST activate a swap
partition before running setup. After making the partition with fdisk,
use: mkswap /dev/<partition> <number of blocks> ; swapon /dev/<partition>
- Once you have prepared the disk partitions for Linux, and activated a
swap partition if you need one, type "setup" to begin the installation
process.
```

```
- If you want the install program to use monochrome displays, type:
TERM=vt100
before you start "setup".

You may now login as "root".

Slackware login:
```

You're now ready to log in to a functioning Linux system. Unlike MS-DOS, UNIX doesn't boot directly to a system prompt. To gain access to UNIX, you must log in and let the system know you're using it. This is similar to how Microsoft Windows for Workgroups and Windows NT behave, except that you can't bypass this step.

At the moment, you don't have an account on your own Linux system. Later in this chapter, you learn how to add a simple user account, whereas Chapter 10, "Managing User Accounts," explains in detail how to manage Linux user accounts. Only one account is available—the one belonging to the *superuser*, also known as *root*. The root account has the highest privileges in the Linux system. When you log in as root, you can do anything you want. So until you add an account for yourself, type **root** and press <Return>.

▶ **See** "Adding a User," **p. 222**

The system responds with

```
Linux 2.0.0. (Posix)

If you're upgrading an existing Slackware system, you might want to remove
old packages before you run 'setup' to install the new ones. If you don't
your system will still work but there might be some old files left laying
around on your drive.

Just mount your Linux partitions under /mnt and type 'pkgtool'. If you
don't know how to mount your partitions, type 'pkgtool' and it will tell
you how it's done.

To start the main installation, type 'setup'

#
```

The # is the superuser prompt. This is similar to the command prompts found under MS-DOS. Linux is waiting for you to issue a command. The preceding information indicates that you should type **setup**, but before you can install the various programs, you must set up your partitions. To do this, you must run the Linux version of fdisk.

Partitioning the Drive for Linux

At the # prompt, enter **fdisk** to start the fdisk program, which displays

```
Using /dev/hda as default device!
```

```
Command (m for help)
```

Notice the first information line printed by fdisk—Using /dev/hda as default device! Remember that MS-DOS refers to most partitions and hard drives as a letter, such as C or D.

Part

I

Ch

4

Linux refers to them in a very different manner. Linux refers to drives as a file name, such as /dev/hda or /dev/hdb.

Linux and MS-DOS communicates with hardware via a series of programs called *device drivers*. Whereas MS-DOS device drivers usually have a .SYS extension and can reside anywhere on the system, Linux stores such device drivers in the /dev directory. The drivers Linux uses in installation were supplied by the boot disk you created. The important point to remember, though, is that because the hard drive, floppy drives, and CD-ROM drives are hardware, Linux uses device drivers in the /dev directory to access the drives. Linux also references these drives with the subdirectory name rather than by a letter. Table 4.6 displays a typical Linux device directory.

Table 4.6 Linux Devices

Device	Name
Floppy drive A	/dev/fd0
Floppy drive B	/dev/fd1
First hard drive	/dev/hda
First primary partition on hard drive A	/dev/hda1
Second primary partition on hard drive A	/dev/hda2
First logical partition on hard drive A	/dev/hda4
Second hard drive	/dev/hdb
First primary partition on hard drive B	/dev/hdb1
First SCSI hard drive	/dev/sda

Notice that the entire hard drive is referred to as /hd*letter*. The primary partitions are then given the next set of four numbers, followed by the logical partitions. Thus, logical partitions always start at /dev/hda4. SCSI hard drives and CD-ROMs follow the same convention, except the hd is replaced by sd.

Using the Linux *fdisk* Program

At the fdisk prompt, enter m for a list of commands. Table 4.7 provides a list of available commands.

Table 4.7 The Linux *fdisk* commands

Command	Description
a	Toggles a bootable flag
c	Toggles the DOS compatibility flag

Command	Description
d	Deletes a partition
l	Lists known partition types
m	Displays this table
n	Adds a new partition
p	Prints the partition table
q	Quits without saving changes
t	Changes a partition's system ID
u	Changes display/entry units
v	Verifies the partition table
w	Writes the table to disk and exits
x	Extra functionality (experts only)

To begin the partitioning, select the p command to display the current partition table, which should reflect the drive you partitioned earlier with the DOS FDISK program. Listing 4.1 shows a possible display from the p command.

Listing 4.1 An Example of a Current Partition Table

```
Disk /dev/hda: 15 heads, 17 sectors, 1024 cylinders
Units = cylinders of 255 * 512 bytes

Device        Boot    Begin    Start     End    Blocks    Id      System
dev/hda2              1024     1024      4040    384667+    51    Novell?
Partition 2 has different physical/logical endings:
phys=(967, 14, 17) Logical=(4096, 14.17)
```

N O T E You may see different information than what's shown here, because the values are different for each type of drive and the partitions already defined on that drive. ▪

The display in Listing 4.1 indicates the various partitions already defined, the start and ending locations of the partition, and how big each partition is in blocks. The display also indicates the partition type. Table 4.8 lists all the different partition types you can define with the Linux fdisk command. The primary partitions you used are 83-Linux Native and 82-Linux Swap. You can get a similar listing with the l command.

Table 4.8 The Known Linux Partition Types

Reference Number	Type
0	Empty
1	DOS 12-bit FAT
2	XENIX root
3	XENIX usr
4	DOS 16-bit < 32M
5	Extended
6	DOS 16-bit >= 32M
7	OS/2 HPFS
8	AIX
9	AIX bootable
a	OS/2 Boot Manager
40	Venix 80286
51	Novell?
52	Microport
63	GNU HURD
64	Novell
75	PC/IX
80	Old MINIX
81	MINIX/Linux
82	Linux Swap
83	Linux Native
93	Amoeba
94	Amoeba BBT
a5	BSD/386
b7	BSDI fs
b8	BSDI swap
c7	Syrinx
db	CP/M
e1	DOS access

Reference Number	Type
e3	DOS R/O
f2	DOS secondary
ff	BBT

Notice the note about the different physical and logical endings at the end of Listing 4.1. The difference is because on the system used to write this chapter, a prior partition containing the DOS D drive was left intact, whereas the C drive was repartitioned to a smaller C drive to make room for Linux. Thus, space is between the C drive and the D drive. This is where the necessary partitions required by Linux will be created.

The begin, start, and end numbers from the display are very important and you should write them down. You'll need them in a later step to specify the necessary sizes of the partitions you'll add.

Adding the Necessary Partition

Because you've repartitioned the drive for MS-DOS, you shouldn't have to delete any partitions for Linux. You only should have to add partitions. To add a partition, issue the n command, which displays

```
Command Action
e extended
p primary(1-4)
```

Press <p> and then <Return>. fdisk asks for the partition number; enter your selection and press <Return>. If you indicate a partition number already in use, fdisk reports this fact and asks you to delete the partition before trying to add it to the partition table. For this example, enter 3 to add a third primary partition that's referred to as /dev/hda3.

Next, fdisk asks for the location of the first cylinder. This is usually the first available cylinder. In fact, fdisk displays a default range for your selection; for example,

```
First cylinder (42-1024) :
```

From this example, you can infer that the first partition ends at cylinder 41 and that the next partition begins at cylinder 1024. Thus, the range supplied by fdisk allows you to start the next partition anywhere in the range of 42 to 1024. It's a very good idea not to place partitions just anywhere throughout the disk, so choose the next available location, which in this case is cylinder 42.

> **N O T E** Linux can have trouble booting from partitions defined to start at cylinders above 1024. If you create a Linux partition starting above 1024, you may have to boot Linux from a floppy. You learn how to create a boot floppy (which is different than the boot floppy used for installation) later in this chapter. The only downside is that booting Linux from a floppy takes a little longer than booting from the hard drive. If you have an IDE drive with more than 1024 cylinders, read the /help/big_ide file on the Slackware 96 CD-ROM. ▪

Part

1

Ch

4

Now fdisk wants you to specify how much space to allocate for this partition. You can express this size in number of cylinders or by the number of bytes (+*size*), kilobytes (+*size*K), or megabytes (+*size*M). Because you should already know the approximate size you need for the swap file, define this partition first, and then leave the rest of the disk space for the Linux program partitions. Thus, for this example, your machine has 8M of RAM, so you need to specify a 16M partition size by replying as such

```
Last cylinder or +size or +sizeM or +sizeK (42-1023): +16M
```

You should then use the p command to look at the new partition table you've defined. In this example, the new partition table looks like

```
Disk /dev/hda: 15 heads, 17 sectors, 1024 cylinders
Units = cylinders of 255 * 512 bytes

Device     Boot Begin Start End   Blocks  Id  System
/dev/hda1   *    1     1    41    5219 1  DOS 12-bit FAT
/dev/hda2        1024  1024 4040  384667+ 51  Novell?

Partition 2 has different physical/logical endings:
phys=(967, 14, 17) Logical=(4039, 14.17)
/dev/hda3         42    42   170   16447+  83    Linux native
```

Note that by default, fdisk made the new partition a Linux native type. To change this to a swap partition, you need to use the t command. Enter **t** and then enter the partition number you want to change; in this example, enter **3**. fdisk then requests that you enter the hexadecimal value of the desired partition type (refer to Table 4.8). If you don't have Table 4.8 handy, you can enter **l** to get the list of partition type codes. Because you want a swap partition, enter **82** at the prompt.

As you can see, fdisk reports the new partition type, but you can also use the p command to double-check that partition 3 is now a Linux swap partition.

Now you can add your Linux partitions. For this example, add only one partition, but if you wanted to have multiple partitions for various reasons, you could do so at this time. To add a partition, press <n> and specify p for another primary partition, and then specify the number for this partition, which is 4. To keep from fragmenting different partitions across the drive, start the last partition where the other left off, at cylinder 171. For the last cylinder, because you want to use the rest of the space for the Linux system, you can specify the last cylinder instead of an exact byte count. Thus, enter **1023**, as follows:

```
Command (m for help):n
Command action
e    extended
p    primary partition (1-4)
p
Partition number (1-4): 4
First cylinder (171-1024):171
Last cylinder or +size or +sizeM or +sizeK (171-1023):1023
```

Now use the p command to verify the new partitions. If you need to make any changes, do so now. When you're satisfied with the layout of your partitions, you can use the w command to write the partition table information to the hard disk.

None of your changes are permanent until you use the w command; thus, if you feel you've made some changes in error, you can use the q command to exit without altering the partition table. If you use the w command, Linux tells you that the partition table has been altered and then resynchronizes the disks to match the new partition table. If your Linux system hangs at this point, reboot with the install boot and root disks until you're back at the # prompt.

Creating the Swap Partition

Some distributions of Slackware, including the one on the accompanying CD-ROM, provide automatic creation and activation of the swap file during installation, so you don't have to worry about creating the swap file. However, if you're using a different distribution version, you may need to create and activate the swap file before continuing with the installation.

▶ **See** "Creating a Swap File," **p. 285**

N O T E   If you get an "out of memory" type error during the following installation procedures, you should increase the size of your swap file. ■

To create the swap space, use the mkswap command and tell the program which partition to use and how much space to use for virtual RAM. For example, to create a swap space on the /dev/hda3 partition that you defined earlier, enter the following command at the # prompt:

```
mkswap -c /dev/hda3 16447
```

The 16447 represents 16M and can be found in the blocks column of the fdisk p command output screen. The optional –c flag tells mkswap to check for bad sections on the partition.

Next, you need to activate the swap system with the swapon command:

```
# swapon /dev/hda3
```

Again, if you're using the accompanying Slackware CD-ROM, you shouldn't have to worry about activating the swap system as long as you create the partition for one swap file. During installation, the install program detects the swap partition and automatically starts the system for installation.

Installing the Linux System

Now that the system is partitioned for Linux, you can install the various packages described in Chapter 2, "Overview of Features." The install package allows you to specify what you want to install. It also detects any DOS partitions and can make them visible to Linux, if you want. To start the installation, type **setup** and press <Return>. Be careful throughout the installation, because the Slackware Setup program doesn't recover well from mistyped or extraneous keys.

If you have a color monitor and have installed the color root disks, you should next see a color display titled Slackware Linux Setup (version FD-2.2.0). The screen provides the menu selections shown in Table 4.9.

Part
I

Ch
4

Table 4.9   The Slackware Linux Setup Screen

Menu Item	Description
HELP	Reads the Slackware Setup Help file
KEYMAP	Remaps your keyboard if you're not using a U.S. keyboard
MAKE TAGS	Experts may customize tagfiles to preselect packages
ADDSWAP	Sets up your swap partition(s)
TARGET	Sets up your target partitions
SOURCE	Selects source media
DISK SETS	Selects disk sets to install
INSTALL	Installs selected disk sets
CONFIGURE	Reconfigures your Linux system
EXIT	Exits Slackware Linux Setup

You can navigate through the menu by using the <↑> and <↓> keys, the <+> and <–> keys, or the key corresponding to the first letter of the menu item previously listed. After you select an option, pressing the space bar or <Return> activates the selection—that is, performs the tasks associated with that menu item. Your first selection should be to review the help file for the Setup program (the HELP menu option). After you review the help file, the recommended sequence for installing Slackware is ADDSWAP, SOURCE, TARGET, DISK SETS, INSTALL, and CONFIGURE.

Setting Up a Swap Partition

Although you can go through the various commands outlined earlier in the section "Creating the Swap Partition" to set up your swap partition, it's easier to allow the Setup program to do this work for you. You can set up and configure your swap partition by selecting the ADDSWAP menu item. This command displays a screen titled SWAP SPACE DETECTED and provides the following information:

```
Slackware Setup has detected a swap partition:

Device    Boot   Begin   Start   End    Blocks   Id   System
/dev/hda3          42      42     170    16447+   82    Linux Swap

Do you wish to install this as your swap partition?
You have two options: yes or no. Choose yes, using the arrow keys, to continue
the installation process. Linux Setup gives the following warning:
IMPORTANT NOTE: If you have already made any of your swap partitions
active (using the swapon command), then you should not allow Setup to
use mkswap on your swap partitions, because it may corrupt memory pages
that are currently swapped out. Instead, you will have to make sure
```

that your swap partitions have been prepared (with mkswap) before they
will work. You might want to do this to any inactive swap partitions
before you reboot.

▶ **See** "Using Swap Files and Partitions," **p. 284**

Choose OK to continue. Setup asks whether you want to use mkswap. If you haven't prepared
the swap files, choose yes and continue. Setup next asks whether you want to make the swap
partitions active with the swapon command. Choose yes to continue the installation. After the
program configures and activates your swap partitions, Setup shows you information that will
be added to the /etc/fstab file.

Choose OK to continue with the installation. Next Setup asks,

```
Now that you've set up your swap space, you may continue on
with the installation. Otherwise, you'll be returned to the
main menu. Would you like to continue the installation and
set up your TARGET drive(s)?
```

Choose yes to set up your TARGET partitions on your hard drive.

Selecting a Target Partition for Installation

Again, the Setup program displays a screen with the partition information. If you've used fdisk
to create multiple partitions, each partition is displayed, and Setup asks you to select one to use
for installation. Because this example uses only one partition, the information for the one parti-
tion is all that's displayed, as shown here:

```
Device     Boot    Begin    Start     End      Blocks    Id     System
/dev/hda4   171     171     1023    108757+    83     Linux native
```

Next, Setup asks you to choose a file system. You have two choices: ext2 and xiafs. Setup
displays the following message and asks for a selection:

```
There are two main filesystem types that are used for Linux.
These are the xiafs filesystem, and the second extended
filesystem (ext2). Ext2 seems to be the current standard.
Xiafs hasn't really been changed in quite some time. Ext2 has one
really nice feature that xiafs doesn't have: an ext2 partition is
unmounted, a clean bit is written to it. When the machine is rebooted,
checking is skipped for any partitions that have the clean bit on them.
Xiafs may be a better choice for machines with low memory, however, so
it's still supported. What filesystem do you plan to use on your root
partition (/dev/hda4 ), ext2fs or ziafs?
```

Installing the ext2 file system is recommended, so choose ext2 and choose OK. You're then
asked to format the partition:

```
If this partition has not been formatted, you should format it.
NOTE: This will erase all data on it. If you are trying to upgrade
an existing Linux partition, you should use setup from your hard drive,
not from the boot/root disk. (The versions of setup supplied on the
hard drive and the boot/root disk differ.) Would you like to format
this partition?
```

Part

I

Ch

4