Caldera®

Hands-On Linux

Mark G. Sobell
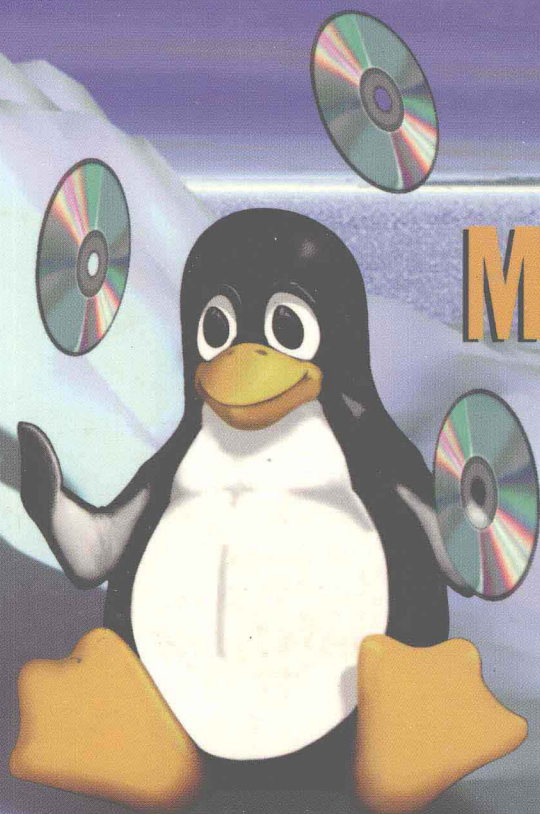
Foreword by
Torvalds

HANDS-ON LINUX

Featuring Caldera® OpenLinux Lite,™ Netscape® Navigator Gold,™ and Netscape FastTrack Server™ on Two CDs

Mark G. Sobell

With much love for my three guys:
Samuel, Zachary, and Max

and

for my wife, Laura,
without whom this book would
never have come to be

Foreword by Linus Torvalds

I got involved with computers around the age of 10 or 11, when my maternal grandfather bought a Commodore VIC-20 (one of the first home machines, at least over here in Finland: 3.5K of RAM, BASIC in ROM, 1MHz 6502 CPU). He got it so early in the lifetime of the VIC that at that time there didn't exist many games or things like that, and I couldn't afford them anyway. So I ended up programming the thing, first in BASIC and then in machine code (not assembler; I didn't have an assembler, so I actually had to learn the numbers rather than symbolic assembly language).

When I went to the university, there was a UNIX course in the fall of '90. That was actually the first time I met UNIX (having used VMS first and hating it). One of the books we studied in that course was *A Practical Guide to the UNIX System* by Mark Sobell. UNIX was really a revelation to me. I had been using other "real" OSs before, but I really *liked* UNIX. So I decided I had to get it for my own machine.

I got a PC, bought Minix for it, and noticed that Minix wasn't really what I wanted (I wanted the *real* thing rather than a small and limited clone). So having programmed for most of my life, I eventually got started on Linux. It wasn't planned; Linux really grew out of another project where I was testing out the features of the 386 chip.

I designed Linux not to be a minimal UNIX and not to be a new operating system. It is a UNIX-like operating system that is close to being source-code compatible with UNIX—meaning you can compile and run UNIX programs under Linux. Linux wouldn't be what it is today without the Internet and the contributions of an incredible number of people. One of the most important and unique facets of the Linux development project has been the effect that feedback (mostly via the Internet) has on development: Feedback accelerates development dramatically.

Several years, lots of work, and many revisions brought Linux version 1.2, which was out in March '95, and had much more stable networking. There were obviously lots of other changes too. By Linux 1.2 I had started the work to port Linux to the DEC Alpha. It worked for me, but Linux/Alpha was by no means stable at that point, and Linux 1.2 was still only useful on x86 PCs.

Version 2.0 was out in June '96. It fully supports the Alpha (64-bit VM) and also has *much* faster networking (1.2 got it stable; 2.0 made it perform well). Linux 2.0 is also SMP-aware, which was the other reason

for the jump in numbering from 1.2 to 2.0 (SMP and multi-architecture: Both are rather large "conceptual" changes). We also improved Linux performance in other areas, so while Linux 1.2 was *good,* 2.0 should be *great.*

As I reflect on the future of Linux, I realize it depends on many things. Linux now has the same problem all other OSs have had: a lack of applications. In a sense this is a good thing: The fact that application availability is the main problem means that Linux has begun to mature and stabilize, and the earlier problems (lack of faith in a new OS, the "hacker" association, etc.) are a thing of the past.

In order to be a *real* driving force, Linux needs to have more applications, and those applications need to be readily available with wide distribution and low price. I've been using ApplixWare, a very nice office suite and a good example of the kind of software I am talking about. The Internet provides a vehicle for wide distribution. So far Linux ports of various applications have followed DOS/Windows pricing (less expensive) rather than the UNIX pricing (more expensive), so I'm reasonably hopeful we can have the types of applications that will help Linux grow.

In five years Linux might well have a noticeable percentage of the PC market. Right now Linux seems to be on the order of 1 percent of PCs, and I don't think it's impossible to have 5 to 10 percent of the market in five years. That doesn't sound like a lot, but it definitely starts to make a difference.

I certainly see Linux used a lot in universities, and we know from observing the growth of UNIX that means that lots of graduates know UNIX/Linux. That is an entree for Linux into business use. Graduates entering the work force will know how easy it is to set up a Linux Web site, mailer daemon, or other application. In fact, I have observed this progression already occurring.

I also have hopes for the growth of Linux in the home market: That's at least as important as the business and educational areas. So I was ecstatic when popular games like Doom and Quake were ported to run on Linux. Games make Linux more appealing to the home market and together with "real" applications, and the Windows and Mac emulation stuff, we have our foot in the door.

Another important factor in boosting Linux awareness and usage in all markets is the availability of books about Linux. Well-written and composed books offer new and experienced Linux users supportive resources beyond the available technical documentation. Books suitable for use in classrooms enable schools to offer courses on Linux, and books on advanced topics allow experienced users to use and apply Linux in many areas. In this book Mark Sobell introduces readers of varying ability to Linux; and once they are using Linux, he provides them with a useful reference for continued use after the first reading. The book provides a bridge to the world of Linux, guiding the beginner and supporting the more experienced user/administrator. Mark's writing style is very accessible, and I like the order he presents the topics in.

It is interesting to reflect on the fact that I initially learned UNIX in part from reading one of Mark's earlier books and now, seven years later, I'm writing a foreword to his new book on Linux. I guess then, in a sense, I am indebted to Mark for helping me to learn UNIX and now for helping to make Linux accessible to more people. This new Linux-specific book offers readers the same accessible style and well-written topic coverage that I found so useful when I first explored UNIX. I strongly recommend it to anyone who is interested in learning and using Linux.

We'll see where Linux goes. We live in interesting times.

L. T.

PREFACE

This book is *practical* because it uses tutorial examples that show you what you will see on your terminal screen each step of the way. It is a *guide* because it takes you from logging in on your system (Chapter 2) through writing complex shell programs (Chapters 11, 12, and 13), using sophisticated software development tools (Chapter 14), and administrating a system (Chapter 15). Part II is a *reference guide* to more than 85 Linux utilities. This *Practical Guide* is intended for people with some computer experience but little or no experience with a Linux/UNIX system. However, more experienced Linux/UNIX system users will find the later chapters and Part II to be useful sources of information on subjects such as GUIs, basic and advanced shell programming, editing, C programming, debugging, source code management, networks, The Internet, The World Wide Web, graphical user interfaces, and Linux system administration.

Audience

This book will appeal to a wide range of readers. As a minimum it assumes some experience with a PC or a Mac, but it does not require any programming experience. It is appropriate for

- Users of both single- and multiuser Linux systems
- Students taking a class about Linux
- Students taking any class in which they use Linux
- Computer science students studying operating systems
- People who want to run Linux at home
- Professionals who want to use Linux at work
- Programmers who need to understand the Linux programming environment

Benefits to You, the Reader

You will come away from this book with a broad knowledge of Linux, and how to use it in day-to-day work. Whether you are a C or Shell programmer or a user who wants to run application programs or use the DOS emulator under Linux, this book will give you the knowledge to proceed. *A Practical Guide to Linux* gives

you a broad understanding of Linux, including how to administer, maintain, and update the system. It will remain a valuable reference tool for years to come.

Scope of Coverage and Features

A Practical Guide to Linux covers a broad range of topics, from writing simple shell scripts to recursive shell programming; from local email to using Netscape to browse the World Wide Web; from using simple utilities to source code management using RCS and CVS; from using a system to administrating one. Below is a list highlighting some of the features of this book, followed by more in-depth discussions of some of these features.

- Compatible with all distributions of Linux
- Broad Internet coverage including Netscape, **ftp**, and downloading software and documentation, using a search engine, and constructing a simple HTML page
- A Help! appendix written in FAQ style that covers everything from setting up special keyboard keys to downloading, compiling, and installing software
- Many examples throughout
- Thorough shell coverage with chapters on the Bourne Again Shell (**bash**), the TC Shell (**tcsh**), and the Z Shell (**zsh**). Coverage includes both interactive use of the shells and programming with the shells
- Using and customizing the X Window System and the **fvwm** window manager
- Using C, **imake**, **make**, and source code management (RCS and CVS) under Linux
- In-depth coverage of the **emacs** and **vi** editors
- Complete instructions on using software from the Internet: finding, downloading, compiling, and installing software from the Internet
- Getting online documentation from many sources (local and Internet)
- A complete discussion of the Linux filesystem
- An appendix covering regular expressions
- A comprehensive index
- An appendix on POSIX standards

The following sections highlight some of the features of this book:

Part I and Part II.   *A Practical Guide to Linux* shows you how to use Linux from your terminal. Part I comprises the first 15 chapters, which contain step-by-step tutorials covering the most important aspects of the Linux operating system. (If you have used a Linux/UNIX system before, you may want to skim over Chapters 2 and 3.) The more advanced material in each chapter is presented in sections marked "Optional," which you are encouraged to return to after mastering the more basic material presented in the chapter. Review exercises are included at the end of each chapter for readers who want to hone their skills. Some of

the exercises test the reader's understanding of material covered in the chapter, while others challenge the reader to go beyond the material presented to develop a more thorough understanding.

Part II offers a comprehensive, detailed reference to the major UNIX utilities, with numerous examples. If you are already familiar with the Linux/UNIX system, this part of the book will be a valuable, easy-to-use reference. If you are not an experienced user, you will find Part II a useful supplement while you are mastering the tutorials in Part I.

Organizing Information. In Chapters 2, 3, and 4, you will learn how to create, delete, copy, move, and search for information using your system. You will also learn how to use the UNIX system file structure to organize the information you store on your computer.

Electronic Mail and Telecommunications. Chapters 2 and 3 and Part II include information on how to use utilities (**pine**, **talk**, and **write**) to communicate with users on your system and other systems. Chapter 7 details how to address electronic mail to users on remote, networked systems.

Using the Shell. In Chapter 5 you will learn how to redirect output from a program to the printer, to your terminal, or to a file—just by changing a command. You will also see how you can use pipes to combine Linux utilities to solve problems right from the command line.

Shell Programming. Once you have mastered the basics of Linux, you can use your knowledge to build more complex and specialized programs using a shell programming language (shell scripts). Chapter 11 shows you how to use the Bourne Again Shell to write your own scripts composed of Linux system commands. Chapter 12 covers the TC Shell. Chapter 13 covers the Z Shell, which combines many of the popular features of the C Shell (such as history and aliases) with a programming language similar to that of the Bourne Shell. This chapter also covers many concepts of advanced shell programming. The examples in Part II also demonstrate many features of the Linux utilities that you can use in shell scripts.

Using Programming Tools. Chapter 14 introduces you to the C compiler and Linux's exceptional programming environment. This chapter describes how to use some of the most useful software development tools: **make**, the Concurrent Versions System (CVS), and the Revision Control System (RCS). The **make** utility automates much of the drudgery involved in ensuring that a program you compile contains the latest versions of all program modules. CVS and RCS help you to track the versions of files involved in a project.

Networking. Chapter 7 is devoted to explaining what a network is, how it works, and how you can use it. It tells you about types of networks, various network implementations, distributed computing, how to use the network for communicating with other users, and using various networking utilities (such as **telnet** and **ftp**).

Internet and the World Wide Web. Chapter 7 also discusses the use of the Internet and shows, with examples, how to use a browser (Netscape) and a search engine (Alta Vista) and how to create a simple page on the Web.

Graphical User Interfaces (GUIs). Chapter 6 discusses the X Window system, how to open and control windows, how to customize your X work environment, and how to customize the Motif and **fvwm** window managers.

The Z Shell and Advanced Shell Programming. Chapter 13 covers many of the features of this powerful shell. It extends the concepts of shell programming introduced in Chapter 11 into more advanced areas, including more information on the locality of variables, recursion, and the coprocess.

The vi **Editor.** The screen-oriented **vi** editor, which was originally a part of Berkeley UNIX, is still one of the most widely used text editors. Chapter 8 starts with a tutorial on **vi** and goes on to explain how to use many of the advanced features of **vi**, including special characters in search strings, the general-purpose and named buffers, parameters, markers, and executing commands from **vi**. The chapter concludes with a summary of **vi** commands.

The emacs **Editor.** Produced and distributed (for minimal cost) by the Free Software Foundation, the **emacs** editor has grown in popularity and is available for Linux. Chapter 9 includes information on **emacs** Version 19 and the X Window System, allowing you to use a mouse and take advantage of X Window System features with **emacs**. This chapter explains how to use many of the features of this versatile editor, from a basic orientation to the use of the META, ALT, and ESCAPE keys; key bindings, buffers, the concept of Point, the cursor, Mark, and Region, incremental and complete searching for both character strings and regular expressions; using the online help facilities, cutting and pasting (from the keyboard and with a mouse), using multiple windows; and C Mode, which is designed to aid a programmer in writing and debugging C code. The chapter concludes with a summary of **emacs** commands.

Job Control. The job control commands, which originated on Berkeley UNIX, allow a user to work on many jobs at once from a single window, and switch back and forth between the jobs as desired. Job control is available under the Bourne Again, TC, and Z shells.

Shell Functions. A feature of the Bourne Again and Z shells, shell functions enable you to write your own commands that are similar to the aliases provided by the TC Shell, only more powerful.

Source Code Management: CVS and RCS. The Concurrent Versions System (CVS) and Revision Control System (RCS) are convenient sets of tools that enable programmers to track multiple versions of files on a number of different types of projects.

POSIX. The IEEE POSIX committees have developed standards for programming and user interfaces based on historical UNIX practice, and new standards are under development. Appendix D describes these standards and their direction and effect on the UNIX industry.

System Administration. Chapter 15 explains the inner workings of the Linux system. It details the responsibilities of the Superuser and explains how to bring up and shut down a Linux system, add users to the system, back up files, set up new devices, check the integrity of a filesystem, and more. This chapter goes into detail about the structure of a filesystem and explains what administrative information is kept in the various files.

Using Linux Utilities. The Linux system includes hundreds of utilities. Part II contains extensive examples of how to use many of these utilities to solve problems without resorting to programming in C (or another language). The example sections of **awk** (over 20 pages, starting on page 648), and **sort** (page 856), give real-life examples that demonstrate how to use these utilities alone and with other utilities to generate reports, summarize data, and extract information.

Regular Expressions.   Many UNIX utilities allow you to use regular expressions to make your job easier. Appendix A explains how to use regular expressions, so that you can take advantage of some of the hidden power of your Linux system.

Supplements

The author's home page (**http://www.sobell.com**) contains downloadable listings of the longer programs from the book; current pointers to many interesting and useful Linux sites on the World Wide Web; a list of corrections to the book; and a solicitation for corrections, comments, suggestions, and additional programs and exercises.

Thanks

Lorraine Callahan and Steve Wampler researched, wrote, analyzed reviews, and coordinated all of the efforts that went into this book. Thank you both very much.

From Pat Parseghian's large-scale system-administration experience at Princeton and her interest in data networks to her work with Linux systems at Transmeta, she brings a breadth to this book that ties together the technobabble of computers and their use in the real world. Pat is responsible for much of the work on the Networking and GUI chapters.

Thanks to the Texan, JFP (Dr. John Frank Peters), for his many hours on the **emacs** chapter. His understanding of this editor gives this chapter a depth and breadth that makes you want to dive right in. Fred Zlotnick, author of *The POSIX.1 Standard,* did a lot of work on the POSIX Appendix.

Larry Ewing (**lewing@isc.tamu.edu**) is responsible for the wonderful penguin playing on the tip of the Linux iceberg on the cover of the book and for other penguins herein as well. He created them with a tool named the GIMP (General Image Manipulation Program—**http://www.isc.tamu.edu/~lewing/gimp**).

Also, a big, "Thank You" to the folks who read through the draft of the book and made comments that caused me to refocus parts of the book where things were not clear or were left out altogether. Thanks to Brian LaRose; Byron A. Jeff, Clark Atlanta University; Charles Stross; Eric H. Herrin, II, University of Kentucky; Jeff Gitlin, Lucent Technologies; Kurt Hockenbury; Maury Bach, Intel Israel Ltd.; Peter H. Salus; Rahul Dave, University of Pennsylvania; Sean Walton, Intelligent Algorithmic Solutions; and Tim Segall, Computer Sciences Corporation for reviewing the book.

A Practical Guide to Linux is based in part on my two previous UNIX books, *A Practical Guide to UNIX* and *UNIX System V: A Practical Guide,* both in their third editions. There were many people who helped me with those books and thanks is due them here: Arnold Robbins, Georgia Tech. University; Behrouz Forouzan, DeAnza College; Mike Keenan, Virginia Polytechnic Institute and State University; Mike Johnson, Oregon State University; Jandelyn Plane, University of Maryland; Sathis Menon, Georgia Tech. University; Cliff Shaffer, Virginia Polytechnic Institute and State University; and Steven Stepanek, California State University, Northridge.

I continue to be grateful to the many people who helped with the early editions of my UNIX books. This book would not have been possible without the help and support of everyone at Informix Software, Inc. Special thanks to Roger Sippl, Laura King, and Roy Harrington for introducing me to the UNIX system. My mother, Dr. Helen Sobell, provided invaluable comments on the manuscript at several junctures.

Dr. Kathleen Hemenway researched, wrote, analyzed reviews, and generally coordinated all the efforts that went into the second edition of my UNIX books. From her work on the UNIX system at Bell Labs, her

teaching experience, and her work at Sun, she brought a breadth to this book that greatly increases its value as a learning tool.

Isaac Rabinovitch provided a very thorough review of the system administration chapter. Professor Raphael Finkel and Professor Randolph Bentson each reviewed the manuscript several times, making many significant improvements. Bob Greenberg, Professor Udo Pooch, Judy Ross, and Dr. Robert Veroff also reviewed the manuscript and made useful suggestions. In addition, the following people provided critical reviews and were generally helpful during the long haul: Dr. Mike Denny, Joe DiMartino, Dr. John Mashey, Diane Schulz, Robert Jung, Charles Whitaker, Don Cragun, Brian Dougherty, Dr. Robert Fish, Guy Harris, Ping Liao, Gary Lindgren, Dr. Jarrett Rosenberg, Dr. Peter Smith, Bill Weber, Mike Bianchi, Scooter Morris, Clarke Echols, Oliver Grillmeyer, Dr. David Korn, Dr. Scott Weikart, and Dr. Richard Curtis.

Dr. Brian Kernighan and Rob Pike graciously allowed me to reprint the **bundle** script from their book, *The UNIX Programming Environment.*

Of course I take responsibility for any errors or omissions. If you find one or just have a comment, let me know (at **mark@sobell.com** or c/o the publisher), and I'll fix it in the next printing. My home page (**http://www.sobell.com**) contains a list of all the errors found so far, and who found them. It also contains copies of the longer scripts from the book and pointers to many interesting Linux pages.

Mark G. Sobell
Menlo Park, California

ABOUT THE CDS

Caldera, Inc., a leading developer of Linux and related packages, has provided the two CD-ROMs at the back of this book. One CD-ROM contains a complete Linux distribution: OpenLinux Lite by Caldera. Except for the Caldera Looking Glass desktop interface, the contents of the OpenLinux Lite CD-ROM are yours to keep. You can use and evaluate Looking Glass for ninety days and then decide if you want to purchase it. The other CD-ROM contains Netscape Navigator Gold and Netscape FastTrack Server, which you can use and evaluate for ninety days and then decide if you want to purchase one or both. This section describes the features of each of these packages and how to install them.

Features of OpenLinux

Following is a list of some key features of OpenLinux Lite. If you are looking for a specific feature, protocol, or system, refer to the documentation on the CD-ROM and on Caldera's Web site. Start with the Caldera Info icon on the desktop or contact Caldera (page xxi).

- Full 32-bit architecture, supporting both ELF and **a.out** binaries
- Multitasking and multiuser capabilities (page 9)
- X Window System distributed graphical environment—X11R6 (page 120)
- Powerful Looking Glass desktop interface (ninety-day demonstration version)
- Graphical text editor—CRiSP-LiTE (**vi** and **emacs** are also included and covered in Chapters 8 and 9)
- Ability to act as a client to other UNIX, Windows NT, and Windows95 systems
- Ability to act as a server to other UNIX, Windows NT, and Windows95 systems
- Remote management via **telnet**, **rlogin**, or with SNMP protocol (included)
- Complete Internet Server Suite, including World Wide Web (Web server software included), FTP, Email (SMTP—**sendmail**), Usenet News (NNTP), **gopher**, **finger**, Telnet Terminal server (host dial-in connections from a modem pool), DNS, NIS, and many others
- Internet Client access via Web browser software

Hardware Requirements	
Processor (Intel) (without FastTrack Server)	386 or better
Processor (Intel) (with FastTrack Server)	486 or better
Hard disk for OpenLinux Lite (recommended install)	>= 300MB
Hard disk for OpenLinux Lite (minimal install)	>= 50MB
CD-ROM drive	
RAM Memory (without the X Window System)	>= 8MB
RAM Memory (with the X Window System)	>= 16MB
RAM Memory (with the FastTrack Server)	>= 32MB
Supported IDE or SCSI hard disk and CD-ROM controller	
Supported IDE or SCSI CD-ROM drive	
Supported graphics controller (for the X Window System)	
A DOS, Linux, or UNIX machine, or a bootable CD-ROM drive (used for installation)	

When you install Linux you will divide the hard disk into two or more partitions. The smaller partition, called the *swap partition,* should be about 30–50MB and the other, called the *Linux filesystem partition,* usually consumes the rest of the disk. Although it is not recommended, you can install Linux without a swap partition if you are tight on disk space. (Performance will suffer.) A minimal install will reduce the disk space required for Linux but many Linux features will not be installed on your hard disk. The X Window System user interface (on the first CD-ROM) allows you to run graphical interfaces including a desktop manager, but it takes up more RAM and hard disk space than a nongraphical implementation. Most hard disk controllers are supported, as are several hundred graphics controllers. Any graphics board will work in low resolution mode and most popular graphics controllers are supported in high resolution mode.

About Caldera OpenLinux

OpenLinux is a complete Linux distribution. It includes many of the standard Linux utilities (similar to the ones found in most other UNIX and UNIX-like operating systems), such as:

- Command line tools
- Text editors, including **vim**—a **vi** clone (Chapter 8), **emacs** (Chapter 9), and **joe** (page 32)
- Programming/development system (Chapter 14)
- UNIX shells **bash**, **tcsh**, and **zsh** (Chapters 10–13)
- X Window System graphical environment (page 120)
- Administrative utilities (Chapter 15)
- The Looking Glass desktop interface (trial version)

- A graphical text editor (CRiSPLiTE)

- A menu-based installation utility (**lisa**)

OpenLinux Lite provides multitasking (page 9) in a multiuser (page 9) environment, using either a graphical or character interface. This Linux system can also take advantage of multiple processor machines (SMP). The kernel features loadable modules that make it quick and easy to add hardware.

Before You Start

If you are new to computers, unfamiliar with this book's conventions, or just want a little help getting started, read the first few pages of Chapter 2.

Installing OpenLinux

If you want more information before you start and you have access to a browser, go to Caldera's Web site (**http://www.caldera.com/doc**) and click on "OpenLinux Base Getting Started Guide." From this page you can read about the installation process in depth. You can start the installation program in one of three ways:

1. If you have a bootable CD-ROM drive, insert the OpenLinux Lite CD-ROM in your machine and reboot.

2. If you have DOS running on your computer, go to the **\Col\Launch\Dos** directory on the CD-ROM. See the **README** file for instructions on how to create the **Loadlin** command line.

3. If option 1 or 2 is not available, you can make the Bootable and Modules diskettes to start the installation. Have two blank, formatted diskettes at hand and follow the installation in the **Readme** file located on the CD-ROM in **\Col\Launch\Floppy**. All the image files you need, and the **Rawrite.exe** program, are also in this directory. You can use either DOS, Linux, or UNIX to prepare the bootable Install and the Module diskettes.

As you start the installation, follow the instructions on the screen. If you need help, try one of the following procedures.

1. Press the FUNCTION-1 key at any time to see a help screen for the choice you are viewing.

2. Press CONTROL, ALT, and FUNCTION-2 at the same time. You will see a login prompt. Log in as help . (If you specified a root password as part of the setup procedure, enter that password. Otherwise you do not need a password.) Choose a menu item from the help catalog. You can switch back to the installation program by pressing CONTROL, ALT, and FUNCTION-1 at the same time.

3. Using your browser, go to Caldera's Web site and view the Getting Started Guide for OpenLinux (**http://www.caldera.com/doc**).

The installation program automatically detects most types of hardware by probing your system. If your hardware does not appear during installation you may need to enter parameters as you start the installation in order to guide the program in locating your hardware correctly. A list of all parameters is provided in the help catalog, which you can view by following the second procedure above.

You can specify the manufacturer and model of a component (for instance, an Ethernet board) by responding *No* to the *second* `Recognized All Hardware?` window. The **lisa** installation program then brings up the Kernel Manager module menu. First choose the type of module (CD-ROM, SCSI adapter, or network card) and then select the component you have. The error message saying `kernel module could not be loaded or initialized` means that **lisa** cannot find the hardware that the requested module supports.

Once your system is installed, you can configure the graphical system using the **XF86Setup** command (if you choose to install XFree86) or **configX** (for the Metro product). In addition, you can run the **lisa** administration utility at any time to view or alter the settings you made during installation. Give the command **lisa**. You can find more information about the X Window System on pages 120 and 139.

After you complete installation by following the on-screen directions, you will be asked if you want to reboot your system. Follow the instructions to do so and log in using the login name `root` or `col` and one of the passwords you specified during installation. *Never* turn your system off without first bringing it down properly. See "Bringing the System Down" on page 593.

The Looking Glass Interface

Caldera OpenLinux Lite comes with the X Window System, a complete graphical environment. Caldera Looking Glass—a desktop manager included on the first CD-ROM—is a tool for file management and system administration. It lets you place icons of commonly used programs and files right on the Desktop background where you have easy access to them. Looking Glass also includes graphical preference settings, a programmable icon bar and icon editor, full file typing with over 1,000 icons, and predefined drag-and-drop functionality.

The Looking Glass interface included with OpenLinux Lite is a trial version. After ninety days, Looking Glass will no longer function. You can continue to use the entire Linux system including the X Window System, but the desktop interface will no longer appear. If you like Looking Glass, see page xx for instructions on how to purchase it. With the X Window System running and **fvwm** on the screen you can use the left mouse button to see a menu of available applications. One of the choices on the `Utilities` menu is `Shells`. Choose one of the shells and X will open an **xterm** window and run the **bash** (Bourne Again) shell (Chapter 10). From this shell you can enter Linux commands and start graphical programs or desktops such as **fvwm** (see pages 130 and 143). Give the command **fvwm** to start this window manager.

Netscape Navigator Gold

The second CD-ROM that accompanies this book includes a ninety day evaluation copy of the Netscape Navigator Gold Internet client. Navigator lets you:

- Browse all Internet Web sites, including sites that use Java. Read and send email (if you have an email server connection through your ISP)

- Read from and post to newsgroups (if you have a newsgroup server connection through your ISP)

- Create HTML documents for Web publishing.

Installing Navigator

There is a discussion of Navigator starting on page 187. The illustrations will not match those you see on the screen, but they will be similar. After installing OpenLinux Lite, log in as **root** and give the following commands:

1. Mount the CD-ROM drive with a command such as the following. The pound sign (#) at the beginning of the line is the **bash** prompt: Do not type this character.

   ```
   # mount /dev/XXX /mnt/cdrom
   ```

   Where **XXX** is the device name of your CD-ROM. If it is a SCSI device the name is probably **scd0**. If it is an IDE device the name will reflect the brand of CD-ROM drive you have. This is the same CD-ROM drive you specified when loading OpenLinux Lite. Give the command **ls /dev/∗cd∗** for a list of CD-ROM drive names.

2. Change directories to the directory the CD-ROM was mounted on in the previous step (give the command **cd /mnt/cdrom** if you used **/mnt/cdrom** in Step 1).

3. Read the **README** file in the working directory (the directory you are in after Step 2). You can use **less** (see "Displaying a Text File" on page 36) to read this file with the command **less README**.

4. After following the pertinent steps in the **README** file, use the following command to extract and install the Netscape Navigator Gold package. Do not type the pound sign.

   ```
   # rpm -i netscape-gold-3.01-5.i386.rpm
   ```

   If you get a message saying `cannot open file` or something similar, give an **ls** command and see if it displays a filename starting with `netscape` and ending with `rpm`. If it does, give the command above substituting the filename you identified for **netscape-gold-3.01-5.i386.rpm**.

After installation, open an **xterm** window in a graphical environment (X Window System and a desktop manager) and give the command **netscape**.

Netscape FastTrack Server

The second CD-ROM also includes a ninety-day evaluation copy of the Netscape FastTrack Server for OpenLinux. The FastTrack Server is a secure Web site, with the ability to encrypt Web traffic and support a commercial Internet site processing credit card transactions or other secure communications. This server includes:

- Support for encrypted transactions using SSL 3.0 (Requires a certificate that you can obtain from a Certificate Authority. This certificate is not included.)

- Support for Java and Javascript browsers

- Ability to use the Netscape server extensions (NSAPI)

- Complete graphical administration of multiple Web servers running FastTrack Server, using a single Netscape Navigator client.