

PREFACE

A Practical Guide to Linux® Commands, Editors, and Shell Programming explains how to work with the Linux operating system from the command line. The first few chapters quickly bring readers with little computer experience up to speed. The rest of the book is appropriate for more experienced computer users. This book does not describe a particular release or distribution of Linux but rather pertains to all recent versions of Linux.

Command line interface (CLI) In the beginning there was the command line (textual) interface (CLI), which enabled you to give Linux commands from the command line. There was no mouse or icons to drag and drop. Some programs, such as emacs, implemented rudimentary windows using the very minimal graphics available in the ASCII character set. Reverse video helped separate areas of the screen. Linux was born and raised in this environment.

Naturally all of the original Linux tools were invoked from the command line. The real power of Linux still lies in this environment, which explains why many Linux professionals work *exclusively* from the command line. Using clear descriptions and lots of examples, this book shows you how to get the most out of your Linux system using the command line interface.

Linux distributions A Linux distribution comprises the Linux kernel, utilities, and application programs. Many distributions are available, including Debian, Red Hat, Fedora Core, SUSE, Mandriva (formerly Mandrake), KNOPPIX, and Slackware. Although the distributions differ from one another in various ways, all of them rely on the Linux kernel, utilities, and applications. This book is based on the code that is common to most distributions. As a consequence you can use it regardless of which distribution you are running.

Overlap If you read *A Practical Guide to Red Hat® Linux®: Fedora Core™ and Red Hat Enterprise Linux, Second Edition*, or a subsequent edition, you will notice some overlap between that book and the one you are reading now. The introduction, the appendix on regular expressions, and the chapters on the utilities (Chapter 3 of this book—*not* Part V), the filesystem, and programming tools are very similar in the two books. The three chapters that cover the Bourne Again Shell (bash) have been expanded and rewritten for this text. Chapters that appear in this book and but not in *A Practical Guide to Red Hat® Linux®, Second Edition*, include those covering the vim and emacs editors, the TC Shell (tcsh), the gawk and sed languages, and Part V, which describes 80 of the most useful Linux utility programs in detail.

Audience This book is designed for a wide range of readers. It does not require programming experience, although some experience using a general-purpose computer is helpful. It is appropriate for the following readers:

- **Students** taking a class in which they use Linux
- **Power users** who want to explore the power of Linux from the command line
- **Professionals** who use Linux at work
- **System administrators** who need a deeper understanding of Linux and the tools that are available to them
- **Computer science students** who are studying the Linux operating system
- **Programmers** who need to understand the Linux programming environment
- **Technical executives** who want to get a grounding in Linux

Benefits *A Practical Guide to Linux® Commands, Editors, and Shell Programming* gives you an in-depth understanding of how to use Linux from the command line. Regardless of your background, it offers the knowledge you need to get on with your work: You will come away from this book understanding how to use Linux, and this text will remain a valuable reference for years to come.

FEATURES OF THIS BOOK

This book is organized for ease of use in different situations. For example, you can read it from cover to cover to learn command line Linux from the ground up. Alternatively, once you are comfortable using Linux, you can use this book as a reference: Look up a topic of interest in the table of contents or index and read about it. Or, refer to one of the utilities covered in Part V, “Command Reference.” You can also think of this book as a catalog of Linux topics: Flip through the pages until a topic catches your eye. The book also includes many pointers to Web sites where you can get additional information: Consider the Web an extension of this book.

A Practical Guide to Linux® Commands, Editors, and Shell Programming offers the following features:

- **Optional sections** allow you to read the book at different levels, returning to more difficult material when you are ready to tackle it.
- **Caution boxes** highlight procedures that can easily go wrong, giving you guidance *before* you run into trouble.
- **Tip boxes** highlight places in the text where you can save time by doing something differently or when it may be useful or just interesting to have additional information.
- **Security boxes** point out ways that you can make a system more secure.
- The **Supporting Web site** at www.sobell.com includes corrections to the book, downloadable examples from the book, pointers to useful Web sites, and answers to even-numbered exercises.
- Concepts are illustrated by **practical examples** found throughout the book.
- The many useful **URLs** (Internet addresses) identify sites where you can obtain software and information.
- **Chapter summaries** review the important points covered in each chapter.
- **Review exercises** are included at the end of each chapter for readers who want to hone their skills. Answers to even-numbered exercises are available at www.sobell.com.
- Important **GNU tools**, including gcc, gdb, GNU Configure and Build System, make, gzip, and many others, are described in detail.
- Pointers throughout the book provide help in obtaining **online documentation** from many sources, including the local system and the Internet.

CONTENTS

This section describes the information that each chapter covers and explains how that information can help you take advantage of the power of Linux. You may want to review the table of contents for more detail.

- **Chapter 1 Welcome to Linux**

Presents background information on Linux. This chapter covers the **history of Linux**, explains how the GNU Project helped Linux get started, and discusses some of **Linux's important features** that distinguish it from other operating systems.

PART I: THE LINUX OPERATING SYSTEM

Experienced users may want to skim Part I

tip If you have used a UNIX/Linux system before, you may want to skim or skip some or all of the chapters in Part I. All readers should take a look at “Conventions Used in This Book” (page 22), which explains the typographic conventions that this book uses, and “Getting the Facts: Where to Find Documentation” (page 29), which points you toward both local and remote sources of Linux documentation.

Part I introduces Linux and gets you started using it.

- **Chapter 2 Getting Started**

Explains the **typographic conventions** that this book uses to make explanations clearer and easier to read. This chapter provides basic information and explains how to log in, **change your password**, give Linux commands using the shell, and **find system documentation**.

- **Chapter 3 Command Line Utilities**

Explains the **command line interface** (CLI) and briefly introduces **more than 30 command line utilities**. Working through this chapter gives you a feel for Linux and introduces some of the tools you will use day in and day out. The utilities covered in this chapter include

- ◆ **grep**, which **searches through files** for strings of characters;
- ◆ **unix2dos**, which **converts Linux text files** to Windows format;
- ◆ **tar**, which **creates archive files** that can hold many other files;
- ◆ **bzip2** and **gzip**, which **compress files** so that they take up less space on disk and allow you to transfer them over a network more quickly; and
- ◆ **diff**, which **displays the differences** between two text files.

- **Chapter 4 The Linux Filesystem**

Discusses the Linux hierarchical filesystem, covering files, filenames, **pathnames**, working with directories, **access permissions**, and hard and **symbolic links**. Understanding the filesystem allows you to **organize your data** so that you can find information quickly. It also enables you to **share some of your files** with other users while **keeping other files private**.

- **Chapter 5 The Shell**

Explains how to use shell features to make your work faster and easier. All of the features covered in this chapter work with both **bash** and **tcsh**. This chapter discusses

- ◆ Using **command line options** to modify the way a command works;
- ◆ How a minor change in a command line can **redirect input** to a command to come from a file instead of the keyboard;
- ◆ How to **redirect output** from a command to go to a file instead of the screen;
- ◆ Using **pipes** to send the output of one utility directly to another utility so that you can solve problems right on the command line;
- ◆ Running programs in the **background** so that you can work on one task while Linux is working on a different one; and
- ◆ Using the shell to **generate filenames** to save you time spent on typing and help you when you do not remember the exact name of a file.

PART II: THE EDITORS

Part II covers two classic, powerful Linux command line text editors. Most Linux distributions include the vim text editor, an “improved” version of the widely used vi editor, as well as the popular GNU emacs editor. Text editors enable you to create and modify text files that can hold programs, shell scripts, memos, and input to text formatting programs. Because Linux system administration involves editing text-based configuration files, skilled Linux administrators are adept at using text editors.

- **Chapter 6 The vim Editor**

Starts with a **tutorial** on vim and then explains how to use many of the **advanced features** of vim, including special characters in search strings, the General-Purpose and Named buffers, parameters, markers, and execution of commands from vim. The chapter concludes with a **summary of vim commands**.

- **Chapter 7 The emacs Editor**

Opens with a **tutorial** and then explains many of the features of the emacs editor as well as how to use the META, ALT, and ESCAPE keys. The chapter also covers key bindings, buffers, and **incremental and complete searching** for both character strings and regular expressions. In addition, it details the relationship between Point, the cursor, Mark, and Region. It also explains how to take advantage of the extensive **online help** facilities available from emacs. Other topics covered include cutting and pasting, using multiple windows and frames, and working with emacs modes—specifically **C mode**, which aids programmers in writing and debugging C code. Chapter 7 concludes with a **summary of emacs commands**.

PART III: THE SHELLS

Part III goes into more detail about bash and introduces the TC Shell (tcsh).

- **Chapter 8 The Bourne Again Shell**

Picks up where Chapter 5 leaves off, covering more advanced aspects of working with a shell. For examples it uses the Bourne Again Shell—bash, the shell used almost exclusively for system shell scripts. Chapter 8 describes how to

- ◆ Use shell **startup files**, shell options, and shell features to **customize your shell**;
- ◆ Use **job control** to stop jobs and move jobs from the foreground to the background and vice versa;
- ◆ Modify and reexecute commands using the **shell history list**;
- ◆ Create **aliases** to customize commands;
- ◆ Work with **user-created and keyword variables** in shell scripts;
- ◆ Set up **functions**, which are similar to shell scripts but can execute more quickly;
- ◆ Write and execute simple **shell scripts**; and
- ◆ **Redirect error messages** so that they go to a file instead of the screen.

- **Chapter 9 The TC Shell**

Describes tcsh and covers features that are common to and different between bash and tcsh. This chapter explains how to

- ◆ Run tcsh and **change your default shell** to tcsh;
- ◆ **Redirect error messages** so that they go to files instead of the screen;
- ◆ Use **control structures** to alter the flow of control within shell scripts;
- ◆ Work with tcsh **array and numeric variables**; and
- ◆ Use shell **builtin commands**.

PART IV: PROGRAMMING TOOLS

Part IV covers programming under Linux. It discusses the C programming environment, the use of bash as a programming language, and ways to write programs using gawk and sed.

- **Chapter 10 Programming Tools**

Introduces Linux's exceptional programming environment. This chapter

- ◆ Explains how to invoke the GNU gcc compiler;
- ◆ Describes how to use make to keep a set of **programs up-to-date**;
- ◆ Explains how to **debug a C program** using gdb;
- ◆ Describes how to work with **shared libraries**;
- ◆ Explains how to set up and use CVS to **manage and track program modules** in a software development project; and
- ◆ Discusses **system calls** and explains how you can use them to initiate kernel operations.

Once you have mastered the basics of Linux, you can use your knowledge to build more complex and specialized programs, using the shell as a programming language.

- **Chapter 11 Programming the Bourne Again Shell**

Shows how to use bash to write advanced shell scripts. This chapter discusses

- ◆ **Control structures** such as **if...then...else** and **case**;
- ◆ **Variables**, including locality of variables;
- ◆ **Arithmetic and logical (Boolean) expressions**; and
- ◆ Some of the most useful **shell builtin commands**, including **exec**, **trap**, and **getopts**.

Chapter 11 poses two complete **shell programming problems** and then shows you how to solve them step by step. The first problem uses **recursion** to create a hierarchy of directories. The second problem develops a quiz program and shows you how to set up a shell script that **interacts with a user** and how the script processes data. (The examples in Part V also demonstrate many features of the utilities you can use in shell scripts.)

- **Chapter 12 The gawk Pattern Processing Language**

Explains how to write programs using the powerful gawk language that filter data, **write reports**, and **retrieve data from the Internet**. The advanced programming section describes how to set up **two-way communication** with another program using a **coprocess** and how to obtain input over a network instead of from a local file.

- **Chapter 13 The sed Editor**

Describes sed, the **noninteractive stream editor** that finds many applications as a filter within shell scripts. This chapter discusses how to use sed's buffers to write **simple yet powerful programs** and includes many examples.

PART V: COMMAND REFERENCE

Linux includes hundreds of utilities. Chapters 11 and 12 as well as Part V provide extensive examples of the use of more than 80 of the **most important utilities** with which you can solve problems without resorting to programming in C. If you are already familiar with UNIX/Linux, this part of the book will be a valuable, **easy-to-use reference**. If you are not an experienced user, it will serve as a useful supplement while you are mastering the earlier sections of the book.

Although the descriptions of the utilities in Chapters 11 and 12 and Part V are presented in a format similar to that used by the Linux manual (man) pages, they are much easier to read and understand. These utilities were chosen because you will work with them **day in and day out** (for example, ls and cp), because they are **powerful tools** that are especially useful in shell scripts (sort, paste, and test), because they help you **work with your Linux system** (ps, kill, and fsck), or because they enable you to **communicate with other systems** (ssh, scp, and ftp). Each utility description includes complete explanations of its most useful options. The “Discussion” and “Notes” sections present **tips and tricks** for using the utility to full advantage. The “Examples” sections demonstrate how to use these utilities in real life, alone and together with other utilities to generate reports, summarize data, and extract information. Take a look at the “Examples” sections for gawk (more than 20 pages, starting on page 537), ftp (page 674), and sort (page 764) to see how extensive these sections are.

PART VI: APPENDIXES

Part VI includes the appendixes, the glossary, and the index.

- **Appendix A Regular Expressions**
Explains how to use **regular expressions** to take advantage of the **hidden power of Linux**. Many utilities, including grep, sed, vim, and gawk, accept regular expressions in place of simple strings of characters. A single regular expression can match many simple strings.
- **Appendix B Help**
Details the steps typically used to **solve the problems** you may encounter with a Linux system. This appendix also includes many **links to Web sites** that offer **documentation**, useful Linux information, mailing lists, and **software**.
- **Appendix C Keeping the System Up-to-date**
Describes how to use tools to download software and **keep your system current**. This appendix includes information on

- ◆ **yum** Downloads software from the Internet, keeping a system up-to-date and **resolving dependencies** as it goes.
- ◆ **Apt** An alternative to yum for keeping a system current.
- ◆ **BitTorrent** Good for distributing large amounts of data such as Linux installation CDs.

- **Glossary**

Defines more than 500 terms that pertain to the use of Linux.

- **Index**

Helps you find the information you want quickly.

SUPPLEMENTS

The author's home page (www.sobell.com) contains downloadable listings of the longer programs from this book as well as pointers to many interesting and useful Linux-related sites on the World Wide Web, a list of corrections to the book, answers to even-numbered exercises, and a solicitation for corrections, comments, and suggestions.

THANKS

First and foremost I want to thank my editor at Prentice Hall PTR, Mark L. Taub, who encouraged me and kept me on track. Mark is unique in my experience: He is an editor who works with the tools I am writing about. Because Mark runs Linux on his home computer, we could share experiences as I wrote. His comments and direction were invaluable. Thank you, Mark T.

A big "Thank You" to the folks who read through the drafts 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: Lars Kellogg-Stedman, Harvard University; Jim A. Lola, Principal Systems Consultant, Privateer Systems, LLC; Eric S. Raymond, cofounder, Open Source Initiative; Scott Mann; Randall Lechlitner, Independent Computer Consultant; Jason Wertz, Computer Science Instructor, Montgomery County Community College; Justin Howell, Solano Community College; Ed Sawicki, The Accelerated Learning Center; David Mercer, Contechst; Jeffrey Bianchine, Advocate, Author, Journalist; John Kennedy; Chris Karr; and Jim Dennis, Starshine Technical Services.

Thanks to Molly Sharp of ContentWorks, the production manager who made sure the book came together as it was supposed to, and to Jill Hobbs, the copy editor who kept the various parts of the English language in their proper relative places. Thanks also to the folks at Prentice Hall PTR who helped bring this book to life: Heather Fox, Publicist; Suzette Ciancio, Marketing Manager; Robin O'Brien, Executive Marketing Manager; Julie Nahil, Full-Service Production Manager; Noreen Regina, Editorial Assistant; and everyone else who worked behind the scenes to make this book happen.

I am also indebted to Denis Howe, the editor of *The Free On-Line Dictionary of Computing* (FOLDOC). Dennis has graciously permitted me to use entries from his compilation. Be sure to visit the dictionary (www.foldoc.org).

Dr. Brian Kernighan and Rob Pike graciously allowed me to reprint the **bundle** script from their book, *The UNIX Programming Environment* (Prentice Hall, 1984).

Parts of *A Practical Guide to Linux® Commands, Editors, and Shell Programming* have grown from my previous Linux books and I want to thank the people who helped with those books.

Thank you to David Chisnall, computer scientist extraordinaire; Carsten Pfeiffer, Software Engineer and KDE Developer; Aaron Weber, Ximian; Matthew Miller, Boston University; Cristof Falk, Software Developer at CritterDesign; Scott Mann, IBM, Systems Management and Integration Professional; Steve Elgersma, Computer Science Department, Princeton University; Scott Dier, University of Minnesota; and Robert Haskins, Computer Net Works.

Thanks also to Dustin Puryear, Puryear Information Technology; Gabor Liptak, Independent Consultant; Bart Schaefer, Chief Technical Officer, iPost; Michael J. Jordan, Web Developer, Linux Online Inc.; Steven Gibson, owner of SuperAnt.com; John Viega, founder and Chief Scientist, Secure Software, Inc.; K. Rachael Treu, Internet Security Analyst, Global Crossing; Kara Pritchard, K & S Pritchard Enterprises, Inc.; Glen Wiley, Capitol One Finances; Karel Baloun, Senior Software Engineer, Looksmart, Ltd.; Matthew Whitworth; Dameon D. Welch-Abernathy, Nokia Systems; Josh Simon, Consultant; Stan Isaacs; and Dr. Eric H. Herrin II, Vice President, Herrin Software Development, Inc. And thanks to Doug Hughes, long-time system designer and administrator, who gave me a big hand with the sections on system administration, networks, the Internet, and programming.

More thanks go to consultants Lorraine Callahan and Steve Wampler; Ronald Hiller, Graburn Technology, Inc.; Charles A. Plater, Wayne State University; Bob Palowoda; Tom Bialaski, Sun Microsystems; Roger Hartmuller, TIS Labs at Network Associates; Kaowen Liu; Andy Spitzer; Rik Schneider; Jesse St. Laurent; Steve Bellenot; Ray W. Hiltbrand; Jennifer Witham; Gert-Jan Hagenaars; and Casper Dik.

A Practical Guide to Linux® Commands, Editors, and Shell Programming is based in part on two of my previous UNIX books: *UNIX System V: A Practical Guide*

and *A Practical Guide to the UNIX System*. Many people helped me with those books, and thanks here go to Pat Parseghian, Dr. Kathleen Hemenway, and Brian LaRose; Byron A. Jeff, Clark Atlanta University; Charles Stross; Jeff Gitlin, Lucent Technologies; Kurt Hockenbury; Maury Bach, Intel Israel Ltd.; Peter H. Salus; Rahul Dave, University of Pennsylvania; Sean Walton, Intelligent Algorithmic Solutions; Tim Segall, Computer Sciences Corporation; Behrouz Forouzan, DeAnza College; Mike Keenan, Virginia Polytechnic Institute and State University; Mike Johnson, Oregon State University; Jandelyn Plane, University of Maryland; Arnold Robbins and Sathis Menon, Georgia Institute of Technology; Cliff Shaffer, Virginia Polytechnic Institute and State University; and Steven Stepanek, California State University, Northridge, for reviewing the book.

I continue to be grateful to the many people who helped with the early editions of my UNIX books. Special thanks are due 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 original manuscript at several junctures. Also, thanks go to Isaac Rabinovitch, Professor Raphael Finkel, Professor Randolph Bentson, Bob Greenberg, Professor Udo Pooch, Judy Ross, Dr. Robert Veroff, 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.

I take responsibility for any errors and omissions in this book. If you find one or just have a comment, let me know (mgs@sobell.com) and I will fix it in the next printing. My home page (www.sobell.com) contains a list of errors and credits those who found them. It also offers copies of the longer scripts from the book and pointers to many interesting Linux pages.

Mark G. Sobell
San Francisco, California

