PREFACE

Linux

A Practical Guide to Linux® Commands, Editors, and Shell Programming, Third Edition, explains how to work with the Linux operating system from the command line. The first few chapters of this book build a foundation for learning about Linux. The rest of the book covers more advanced topics and goes into more detail. This book does not describe a particular release or distribution of Linux but rather pertains to all recent versions of Linux.

Mac OS X

This book also explains how to work with the UNIX/Linux foundation of Mac OS X. It looks "under the hood," past the traditional GUI (graphical user interface) that most people associate with the Macintosh, and explains how to use the powerful command-line interface (CLI) that connects you directly to OS X. Where this book refers to Linux, it implicitly refers to Mac OS X as well and makes note of differences between the two operating systems.

Command-line interface (CLI)

In the beginning there was the command-line (textual) interface, which enabled a user to give Linux commands from the command line. There was no mouse to point with 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, so naturally all 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 many 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 Ubuntu, Fedora, openSUSE, Red Hat, Debian, Mageia, Arch, CentOS, and Mint. 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.

New in this edition

This edition includes a wealth of new and updated material:

- A new chapter on the Python programming language (Chapter 12).
- A new chapter on the MySQL database management system (Chapter 13).
- A new chapter on the OpenSSH secure communication utilities including ssh (Chapter 17).
- Rewritten chapters on bash (Chapters 5, 8, and 10) that include new information on
  - The hash table and the hash builtin;
  - Using brace expansion with a sequence expression to generate step values;
  - Determining whether a file descriptor is associated with the terminal;
  - Environment, environment versus local variables, inheritance, and process locality;
  - Using eval to scan, evaluate, and execute a command line;
  - Control operators and implicit command-line continuation;
  - Localization and internationalization including discussions of the locale utility and the **LC_** variables; and
  - Setting the time zone, the **TZ** environment variable, and the tzselect utility.
- Many new command-line utilities in Part VI, including
  - screen, a terminal session manager/multiplexer;
  - busybox, a single binary collection of utilities;
  - The Midnight Commander (mc) file manager;
  - sshfs/curlftpfs, which allows you to mount a directory on an OpenSSH or FTP server as a local directory;
  - expand/unexpand, which converts TABs to SPACEs and vice versa;
  - join, which joins lines from two files based on a common field;
  - nl, which numbers lines from a file; and
  - printf, which formats string and numeric data.

Overlap

If you read one of Mark Sobell's other books, *A Practical Guide to Fedora™ and Red Hat® Enterprise Linux, Sixth Edition,* or *A Practical Guide to Ubuntu Linux®, Third Edition,* or a subsequent edition of either book, you will notice some overlap between

those books 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 VI), the filesystem, the Bourne Again Shell (bash), and Perl are very similar in the books. Chapters that appear in this book but not in the other two books include those covering the vim and emacs editors, the TC Shell (tcsh), the AWK and sed languages, the Python programming language, the MySQL RDBMS, the rsync utility, and Part VI, which describes 98 of the most useful Linux and Mac OS X 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 computer is helpful. It is appropriate for the following readers:

- **Students** taking a class in which they use Linux or Mac OS X
- **Power users** who want to explore the power of Linux or Mac OS X from the command line
- **Professionals** who use Linux or Mac OS X at work
- **Beginning Macintosh users** who want to know what UNIX/Linux is, why everyone keeps saying it is important, and how to take advantage of it
- **Experienced Macintosh users** who want to know how to take advantage of the power of UNIX/Linux that underlies Mac OS X
- **UNIX users** who want to adapt their UNIX skills to the Linux or Mac OS X environment
- **System administrators** who need a deeper understanding of Linux or Mac OS X and the tools that are available to them, including the bash, Perl, and Python scripting languages
- **Web developers** who need to understand Linux inside and out including Perl and Python
- **Computer science students** who are studying the Linux or Mac OS X operating system
- **Programmers** who need to understand the Linux or Mac OS X programming environment
- **Technical executives** who want to get a grounding in Linux or Mac OS X

Benefits

A Practical Guide to Linux® Commands, Editors, and Shell Programming, Third Edition, gives you a broad understanding of how to use Linux and Mac OS X 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 with an understanding of how to use Linux/OS X, and this text will remain a valuable reference for years to come.

A large amount of free software has always been available for Macintosh systems. In addition, the Macintosh shareware community is very active. By introducing the UNIX/Linux aspects of Mac OS X, this book throws open to Macintosh users the vast store of free and low-cost software available for Linux and other UNIX-like systems.

In this book, *Linux* refers to *Linux* and *Mac OS X*

tip The UNIX operating system is the common ancestor of Linux and Mac OS X. Although the GUIs (graphical user interfaces) of these two operating systems differ significantly, the command-line interfaces (CLIs) are very similar and in many cases identical. This book describes the CLIs of both Linux and Mac OS X. To make the content more readable, this book uses the term *Linux* to refer to both *Linux* and *Mac OS X*. It makes explicit note of where the two operating systems differ.

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 VI, "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 obtain additional information: Consider the Internet to be an extension of this book.

A Practical Guide to Linux® Commands, Editors, and Shell Programming, Third Edition, 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 might be useful or just interesting to have additional information.
- **Security boxes** point out ways you can make a system more secure.
- Each chapter starts with a list of **chapter objectives**—a list of important tasks you should be able to perform after reading the chapter.
- 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.
- **Main, File Tree, and Utility indexes** help you find what you are looking for quickly; for easy access, the Utility index is reproduced on the insides of the front and back covers.
- **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 posted at www.sobell.com.

- Important **GNU tools**, including gcc, 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.
- Important command-line utilities that were developed by Apple specifically for Mac OS X are covered in detail, including diskutil, ditto, dscl, GetFileInfo, launchctl, otool, plutil, and SetFile.
- Descriptions of Mac OS X **extended attributes** include **file forks**, **file attributes**, **attribute flags**, and **Access Control Lists** (**ACLs**).
- Appendix D, "Mac OS X Notes," lists some differences between Mac OS X and Linux.

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 might want to review the table of contents for more detail.

- **Chapter 1—Welcome to Linux and Mac OS X**
  Presents background information on Linux and OS X. This chapter covers the **history of Linux**, profiles the OS X **Mach kernel**, explains how the GNU Project helped Linux get started, and discusses some of the **important features of Linux** that distinguish it from other operating systems.

PART I: THE LINUX AND MAC OS X OPERATING SYSTEMS

Experienced users might want to skim Part I

tip If you have used a UNIX/Linux system before, you might 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 24), which explains the typographic conventions that this book uses, and "Where to Find Documentation" (page 33), 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** 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—The 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, day out. Deeper discussion of utilities is reserved for Part VI. 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 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;
  - Making minor changes in a command line to **redirect input** to a command so that it comes from a file instead of the keyboard;
  - **Redirecting output** from a command to go to a file instead of the screen;
  - Using **pipelines** to send the output of one utility directly to another utility so you can solve problems right on the command line;
  - Running programs in the **background** so you can work on one task while Linux is working on a different one; and
  - Using the shell to **generate filenames** to save 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 within 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. In addition, this chapter covers key bindings, buffers, and **incremental and complete searching** for both character strings and regular expressions. 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** (bash)
  Picks up where Chapter 5 left 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 the 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;
  - Implement localization including discussions of the locale utility, the **LC_** variables, and internationalization;
  - Set up **functions**, which are similar to shell scripts but are executed more quickly;
  - Write and execute simple **shell scripts**; and
  - **Redirect error messages** so they go to a file instead of the screen.

- **Chapter 9—The TC Shell** (tcsh)
  Describes tcsh and covers features 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 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 important programming tools that are used extensively in Linux and Mac OS X system administration and general-purpose programming.

- **Chapter 10—Programming the Bourne Again Shell** (bash)
  Continues where Chapter 8 left off, going into greater depth about advanced shell programming using bash, with the discussion enhanced by extensive examples. This chapter discusses
  - **Control structures** including **if...then...else** and **case**;
  - **Variables**, with discussions of attributes, expanding null and unset variables, array variables, and variables in functions;
  - **Environment**, including environment versus local variables, inheritance, and process locality;
  - **Arithmetic and logical** (**Boolean**) **expressions**; and
  - Some of the most useful **shell builtin commands**, including exec, trap, and getopts.

  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 10 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, shows you how to set up a shell script that **interacts with a user**, and explains how the script processes data. (The examples in Part VI also demonstrate many features of the utilities you can use in shell scripts.)

- **Chapter 11—The Perl Scripting Language**
  Introduces the popular, feature-rich Perl programming language. This chapter covers
  - Perl **help tools** including perldoc;
  - Perl **variables** and **control structures**;

- **File handling**;
- **Regular expressions**; and
- Installation and use of **CPAN modules**.

Many Linux administration scripts are written in Perl. After reading Chapter 11 you will be able to better understand these scripts and start writing your own. This chapter includes many examples of Perl scripts.

- **Chapter 12—The Python Programming Language**
  Introduces the flexible and friendly Python programming language. This chapter covers
  - Python **lists** and **dictionaries**;
  - Python functions and methods you can use to **write to and read from files**;
  - Using **pickle** to store an object on disk;
  - Importing and using **libraries**;
  - Defining and using **functions**, including regular and **Lambda functions**;
  - **Regular expressions**; and
  - Using **list comprehensions**.

  Many Linux tools are written in Python. Chapter 12 introduces Python, including some basic object-oriented concepts, so you can read and understand Python programs and write your own. This chapter includes many examples of Python programs.
- **Chapter 13—The MySQL Database Management System**
  Introduces the widely used MySQL relational database management system (RDBMS). This chapter covers
  - Relational database **terminology**;
  - **Installing** the MySQL client and server;
  - **Creating a database**;
  - **Adding a user**;
  - Creating and modifying **tables**;
  - **Adding data** to a database; and
  - **Backing up** and restoring a database.
- **Chapter 14—The AWK Pattern Processing Language**
  Explains how to use the powerful AWK language to write programs 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 15—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: SECURE NETWORK UTILITIES

Part V describes two utilities you can use to work on a remote system and copy files across a network securely.

- **Chapter 16—The rsync Secure Copy Utility**
  Covers rsync, a secure utility that **copies** an ordinary file or **directory hierarchy** locally or between the local system and **a remote system**. As you write programs, you can use this utility to back them up to another system.
- **Chapter 17—The OpenSSH Secure Communication Utilities**
  Explains how to use the ssh, scp, and sftp utilities to communicate securely over the Internet. This chapter covers the use of authorized keys that allow you to log in on a remote system securely without a password, ssh-agent that can hold your **private keys** while you are working, and **forwarding X11** so you can run graphical programs remotely.

PART VI: COMMAND REFERENCE

Linux includes hundreds of utilities. Chapters 14, 15, 16, and 17 as well as Part VI provide extensive examples of the use of over 100 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 14, 15, 16, and 17 and Part VI 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 are included because you will work with them **day in, 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 a 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, differentiating between options supported under Mac OS X and those supported under Linux. The "Discussion" and "Notes" sections present **tips and tricks** for taking full advantage of the utility's power. 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 find (page 808), ftp (page 822), and sort (page 944) to see how extensive these sections are. Some utilities, such as Midnight Commander (mc; page 879) and screen (page 931), include extensive discussion sections and tutorials.

PART VII: APPENDIXES

Part VII includes the appendixes, the glossary, and three indexes.

- **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, AWK, Perl, and Python, 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 might encounter when using a Linux system.
- **Appendix C—Keeping the System Up-to-Date**
  Describes how to use tools to download software and **keep a 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-get—An alternative to yum for keeping a system current.
  - **BitTorrent**—Good for distributing large amounts of data such as Linux installation CDs and DVDs.
- **Appendix D—Mac OS X Notes**
  A brief guide to Mac OS X features and quirks that might be unfamiliar to users who have been using Linux or other UNIX-like systems.
- **Glossary**
  Defines more than 500 terms that pertain to the use of Linux and Mac OS X.
- **Indexes**
  Three indexes that make it easier to find what you are looking for quickly. These indexes indicate where you can locate tables (page numbers followed by the letter t) and definitions (*italic* page numbers). They also differentiate between light and comprehensive coverage (page numbers in light and standard fonts, respectively).
  - **File Tree Index**—Lists, in hierarchical fashion, most files mentioned in this book. These files are also listed in the Main index.
  - **Utility Index**—Locates all utilities mentioned in this book. A page number in a light font indicates a brief mention of the utility; use of the regular font indicates more substantial coverage. The Utility index is reproduced on the insides of the front and back covers.
  - **Main 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- and OS X-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.

PowerPoint slides of the figures in this book and answers to end-of-chapter exercises are provided to qualified instructors for classroom use. If you want to obtain these ancillaries, contact your Pearson representative.

THANKS

First and foremost, I want to thank Mark L. Taub, Editor-in-Chief, Prentice Hall, who provided encouragement and support through the hard parts of this project. Mark is unique in my 30 years of book writing experience: an editor who works with the tools I write about. Because Mark runs Linux on his home computer, we shared experiences as I wrote this book. Mark, your comments and direction are invaluable; this book would not exist without your help. Thank you, Mark T.

The production people at Prentice Hall are wonderful to work with: Julie Nahil, Full-Service Production Manager, worked with me day-by-day during production of this book providing help and keeping everything on track, while John Fuller, Managing Editor, kept the large view in focus. Thanks to Jill Hobbs, Copyeditor; and Audrey Doyle, Proofreader, who made each page sparkle and found the mistakes I left behind.

Thanks also to the folks at Prentice Hall who helped bring this book to life, especially Kim Boedigheimer, Editorial Assistant, who attended to the many details involved in publishing this book; Heather Fox, Publicist; Stephane Nakib, Marketing Manager; Cheryl Lenser, Senior Indexer; Sandra Schroeder, Design Manager; Chuti Prasertsith, Cover Designer; and everyone else who worked behind the scenes to make this book come into being.

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

Special thanks go to Max Sobell, Intrepidus Group, for his extensive help writing the Python chapter; Doug Hellmann, Senior Developer, DreamHost, for his careful and insightful reviews of the Python chapter; and Angjoo Kanazawa, Graduate Student, University of Maryland, College Park, for her helpful comments on this chapter.

Thanks to Graham Lee, Mobile App Developer and Software Security Consultant, Agant, Ltd., and David Chisnall, University of Cambridge, for their reviews and comments on the Mac-related sections of this book.

In his reviews, Jeffrey S. Haemer taught me many tricks of the bash trade. I had no idea how many ways you could get bash to do your bidding. Jeffrey, you are a master; thank you for your help.

In addition to her insightful comments on many sections, Jennifer Davis, Yahoo! Sherpa Service Engineering Team Lead, used her thorough understanding of MySQL to cause me to change many aspects of that chapter.

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: Michael Karpeles; Robert P. J. Day, Candy Strategies; Gavin Knight, Noisebridge; Susan Lauber, Lauber System Solutions, Inc.; William Skiba; Carlton "Cobolt" Sue; Rickard Körkkö, Bolero AB; and Benjamin Schupak.

Thanks also to the following people who helped with my previous Linux books, which provided a foundation for this book:

Doug Hughes; Richard Woodbury, Site Reliability Engineer, Google; Max Sobell, Intrepidus Group; Lennart Poettering, Red Hat, Inc.; George Vish II, Senior Education Consultant, Hewlett-Packard; Matthew Miller, Senior Systems Analyst/Administrator, BU Linux Project, Boston University Office of Information Technology; Garth Snyder; Nathan Handler; Dick Seabrook, Emeritus Professor, Anne Arundel Community College; Chris Karr, Audacious Software; Scott McCrea, Instructor, ITT Technical Schools; John Dong, Ubuntu Developer, Forums Council Member; Andy Lester, author of *Land the Tech Job You Love: Why Skill and Luck Are Not Enough;* Scott James Remnant, Ubuntu Development Manager and Desktop Team Leader; David Chisnall, Swansea University; Scott Mann, Aztek Networks; Thomas Achtemichuk, Mansueto Ventures; Daniel R. Arfsten, Pro/Engineer Drafter/Designer; Chris Cooper, Senior Education Consultant, Hewlett-Packard Education Services; Sameer Verma, Associate Professor of Information Systems, San Francisco State University; Valerie Chau, Palomar College and Programmers Guild; James Kratzer; Sean McAllister; Nathan Eckenrode, New York Ubuntu Local Community Team; Christer Edwards; Nicolas Merline; Michael Price; Mike Basinger, Ubuntu Community and Forums Council Member; Joe Barker, Ubuntu Forums Staff Member; James Stockford, Systemateka, Inc.; Stephanie Troeth, Book Oven; Doug Sheppard; Bryan Helvey, IT Director, OpenGeoSolutions; and Vann Scott, Baker College of Flint.

Also, thanks to Jesse Keating, Fedora Project; Carsten Pfeiffer, Software Engineer and KDE Developer; Aaron Weber, Ximian; Cristof Falk, Software Developer, CritterDesign; Steve Elgersma, Computer Science Department, Princeton University; Scott Dier, University of Minnesota; Robert Haskins, Computer Net Works; Lars Kellogg-Stedman, Harvard University; Jim A. Lola, Principal Systems Consultant, Privateer Systems; 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; Jeffrey Bianchine, Advocate, Author, Journalist; John Kennedy; and Jim Dennis, Starshine Technical Services.

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; Steven Gibson, Owner, SuperAnt.com; John Viega, Founder and Chief Scientist, Secure Software; K. Rachael Treu, Internet

Security Analyst, Global Crossing; Kara Pritchard, K & S Pritchard Enterprises; Glen Wiley, Capital One Finances; Karel Baloun, Senior Software Engineer, Looksmart; Matthew Whitworth; Dameon D. Welch-Abernathy, Nokia Systems; Josh Simon, Consultant; Stan Isaacs; and Dr. Eric H. Herrin II, Vice President, Herrin Software Development.

More thanks go to consultants Lorraine Callahan and Steve Wampler; Ronald Hiller, Graburn Technology; 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, Third Edition, 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; Brian LaRose; Byron A. Jeff, Clark Atlanta University; Charles Stross; Jeff Gitlin, Lucent Technologies; Kurt Hockenbury; Maury Bach, Intel Israel; 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 this 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 interesting Linux pages on the Internet. You can follow me at twitter.com/marksobell.

Mark G. Sobell
San Francisco, California