

FOURTH EDITION

“First Sobell taught people how to use Linux... now he teaches you the power of Linux. A must-have book for anyone who wants to take Linux to the next level.”

—Jon “maddog” Hall, Executive Director, Linux International

A Practical Guide to Linux[®] Commands, Editors, and Shell Programming

Discover the Power of Linux —
Covers macOS, too!

- ▶ Learn from hundreds of realistic, high-quality examples, and become a true command-line guru
- ▶ Covers MariaDB, DNF, and Python 3
- ▶ 300+ page reference section covers 102 utilities, including macOS commands

For use with
all popular versions of
Linux, including Ubuntu,[™]
Fedora,[™] openSUSE,[™]
Red Hat,[®] Debian, Mageia,
Mint, Arch, CentOS,
and macOS

Mark G. Sobell
coauthored by **Matthew Helmke**



UTILITY INDEX

A light page number such as 456 indicates a brief mention. Page numbers followed by the letter **t** refer to tables.

SYMBOLS

: (null) 486, 498
.(dot) 291, 493
[[...]] 506
@ 396, 398, 400, 418
% 418

A alias 352, 387, 418
alloc 418
apropos 35
apt-get 1060–1064
ash *see* dash
aspell 449, 739–742
at 743–746
atq 743–746
atrm 743–746
atrun (macOS) 744
autoconf 771
awk *see* mawk

B basename 463
bash *see* bash in
the Main index
(page 1141)
batch 743–746
bg 151, 306, 418
bind 350
bindkey 393, 394, 418
builtins 419, 504t
bunzip2 65, 750–751
busybox 747–749
bzipcat 65, 750–751
bzip2 64, 750–751
bzip2recover 65,
750–751

C cal 752
cat 52, 138, 141, 145,
436, 710, 753–754
cd 94, 117, 323, 419,
755–756

chdir 419
chgrp 121, 757–758
chmod 102–104, 295,
756–763
chmod (macOS)
1074–1076
chown 764–765
chsh 287, 381
cmp 766–767
col 900
comm 768–769
compress 65, 86, 859,
997, 997
configure 770–771
cp 53, 97, 131,
772–775
cpio 776–780, 823, 826,
860
cron 781–783
crontab 781–783
curlftpfs 983–985
cut 369, 784–785

D dash 287
date 62, 329, 330, 331,
383, 472, 787–789
dd 711, 790–792
declare 315–316, 487
df 793–794
diff 59, 795–799
diff3 796
dircolors 888
dirs 307, 389, 419
disktool (macOS) 801
diskutil (macOS)
800–802
ditto (macOS) 803–804
dmesg 805
dos2unix 63
dscl (macOS) 806–808
du 809–811

E e2fsck 835
echo 61, 154, 381, 419,
457, 458, 476, 812–813
ed 56, 166, 795
egrep 853, 854, 1044
elvis 166
emacs 222–278
env 358, 483
eval 358, 419, 500
ex 166, 175
exec 416, 419, 465,
493–496
exit 382, 407, 419, 434,
477
expand 814–815
export 316, 481–482
expr 816–819

F false 302
fc 338–340
fg 151, 305, 419
fgrep 853, 854
file 60, 332, 820–821
filetest 411, 419
find 390, 442, 822–827
finger 72, 828–829
fmt 831–832
free 74
fromdos 63
fsck 833–837
fsck (macOS) 801
ftp 838–843
fusermount 981–983

G g++ 850–854
gawk 447, 636–668, 1044
gcc 846–850
gdb 847
getfacl 108–112
GetFileInfo (macOS)
851–852

getopts 501–503
getty 333
glob 420
grep 56, 148, 450, 462,
531, 827, 853–857
gunzip 66, 858–860, 997
gzip 66, 858–860, 997

H hash 336
hashstat 420
head 57, 861–862
history 336, 337, 384,
420
hostname 53

I info 36–38, 158
ispell *see* aspell

J jobs 30, 152, 305, 420
join 863–865

K kill 30, 152, 424, 496,
499, 866–867, 949
killall 868–869

L launchctl (macOS)
745, 870–871
ldd 928
less 34, 53, 148, 436,
873–876
let 370, 505
lftp 715
limit 420
ln 113, 116, 878–880
local 357, 488
locale 328–330
locate 70
log 420
login 333, 420
logout 382, 407, 420
lpq 56, 881–883

Continued on inside back cover

UTILITY INDEX

Continued from inside front cover

- lpr 55, 147, 881–883
 - lprm 56, 881–883
 - lpstat 55
 - ls 52, 100, 119, 884–890
 - ls-F 407, 420
- M**
- mac2unix 63
 - make 569, 892–897
 - man 33–35, 898–901
 - mawk *see* gawk
 - mc 902–908
 - mesg 76
 - mingetty 333
 - mkdir 93–94, 909
 - mkfs 911–913
 - mklost+found 836
 - mlocate 70
 - more 53, 874
 - mount 107
 - mv 54, 97, 98, 914–915
 - mysqldump 625
- N**
- ncal 752
 - nice 420, 916–917
 - nl 918–919
 - nohup 420, 920
 - notify 420
- O**
- od 921–924
 - onintr 412, 420
 - open (macOS) 926–927
 - otool (macOS) 928–929
- P**
- paste 930–931
 - pax 932–937
 - perldoc 531
 - pinfo 37
 - plutil (macOS) 938–939
 - popd 309, 420
 - port 1077
 - pr 940–941
 - printenv 420, 482
 - printf 942–945
 - ps 152, 298, 334, 476, 946–950
 - ps2pdf 900
 - pstree 334
 - pushd 308, 408, 421
 - pwck 660
 - pwd 89, 117, 129
 - pwgen 45
- R**
- read 451, 489–492, 492–493
 - readlink 120
 - readonly 315, 316
 - rehash 336, 421
 - renice 951
 - repeat 421
 - rm 52, 118, 408, 953–954
 - rmdir 96, 955
 - rsync 690–700, 714
 - rtorrent 1065
 - run-parts 782
- S**
- sched 421
 - scp 707, 713–714 *see also* OpenSSH in the Main index (page 1141)
 - screen 958–963
 - script 62
 - sdiff 796, 797
 - sed 670–686
 - seq 367, 445
 - set 360, 396, 400, 421, 439, 442, 472, 484
 - setenv 396, 421, 481
 - setfacl 108–112
 - SetFile (macOS) 965–966
 - sftp 715
 - sh 287, 1086
 - shift 421, 436, 473
 - shopt 360
 - shred 954
 - sleep 477, 967–968
 - sort 58, 147, 374, 466, 969–977
 - source 291, 421
- T**
- tac 683, 753
 - tail 57, 390, 992–994
 - tar 66–68, 303, 995–999
 - tee 149, 1000
 - telnet 1001–1002
 - Terminal (macOS) 1076
 - test 431–433, 434, 438, 441, 444, 448, 453, 468–469, 1005–1007
 - time 405, 421
 - todos 63
 - top 1008–1010
 - touch 96, 143, 751, 1012–1013
 - tput 452
 - tr 64, 146, 292, 785, 1014–1016
 - trap 452, 496–499
 - true 302, 498
 - tty 1017
 - tune2fs 1018–1019, 1020
 - type 489
- U**
- split 978–979
 - ssh 27, 707, 709–713 *see also* OpenSSH in the Main index (page 1141)
 - ssh-add 721
 - ssh-agent 720–721
 - ssh-copy-id 719
 - sshfs 981–983
 - ssh-import-id 705
 - ssh-keygen 718–720
 - stat 984–985
 - stop 421
 - strings 986
 - stty 29, 987–990
 - su 32
 - sudo 33
 - suspend 421
 - sysctl (macOS) 991
 - systemsetup (macOS) 332
 - typeset *see* declare
 - tzconfig 331
 - tzselect 331
- U**
- umask 422, 1021–1022
 - unalias 353, 355, 387, 422
 - uname 709
 - uncompress 86, 997
 - unexpand 814–815
 - unhash 422
 - uniq 58, 1023–1024
 - unix2dos 63
 - unix2mac 63
 - unlimit 422
 - unset 314, 422
 - unsetenv 397, 422
 - updatedb 70
 - uptime 73, 1008, 1025
- V**
- vi 166
 - view 177
 - vile 166
 - vim 166–217
 - vimtutor 165
- W**
- w 73, 1025–1026
 - wait 422
 - wc 61, 396, 1027
 - whatis 36
 - where 422
 - whereis 69
 - which 69, 422, 899, 1028–1029
 - who 71, 1030–1031
 - write 75, 1030
- X**
- xargs 1032–1034
- Y**
- yum 1054–1060
 - yumdownloader 1058
- Z**
- zcat 66, 858–860
 - zdiff 860
 - zless 860

PRAISE FOR PREVIOUS EDITIONS OF *A PRACTICAL GUIDE TO LINUX[®] COMMANDS, EDITORS, AND SHELL PROGRAMMING*

“This book is a very useful tool for anyone who wants to ‘look under the hood’ so to speak, and really start putting the power of Linux to work. What I find particularly frustrating about man pages is that they never include examples. Sobell, on the other hand, outlines very clearly what the command does and then gives several common, easy-to-understand examples that make it a breeze to start shell programming on one’s own. As with Sobell’s other works, this is simple, straight-forward, and easy to read. It’s a great book and will stay on the shelf at easy arm’s reach for a long time.”

—*Ray Bartlett*
Travel Writer

“Overall I found this book to be quite excellent, and it has earned a spot on the very front of my bookshelf. It covers the real ‘guts’ of Linux— the command line and its utilities—and does so very well. Its strongest points are the outstanding use of examples, and the Command Reference section. Highly recommended for Linux users of all skill levels. Well done to Mark Sobell and Prentice Hall for this outstanding book!”

—*Dan Clough*
Electronics Engineer and
Slackware Linux User

“Totally unlike most Linux books, this book avoids discussing everything via GUI and jumps right into making the power of the command line your friend.”

—*Bjorn Tipling*
Software Engineer
ask.com

“This book is the best distro-agnostic, foundational Linux reference I’ve ever seen, out of dozens of Linux-related books I’ve read. Finding this book was a real stroke of luck. If you want to really understand how to get things done at the command line, where the power and flexibility of free UNIX-like OSes really live, this book is among the best tools you’ll find toward that end.”

—*Chad Perrin*
Writer, TechRepublic

“I moved to Linux from Windows XP a couple of years ago, and after some distro hopping settled on Linux Mint. At age 69 I thought I might be biting off more than I could chew, but thanks to much reading and the help of a local LUG I am now quite at home with Linux at the GUI level.

“Now I want to learn more about the CLI and a few months ago bought your book: *A Practical Guide to Linux® Commands, Editors, and Shell Programming, Second Edition*.

“For me, this book is proving to be the foundation upon which my understanding of the CLI is being built. As a comparative ‘newbie’ to the Linux world, I find your book a wonderful, easy-to-follow guide that I highly recommend to other Linux users.”

—John Nawell
CQLUG (Central Queensland
Linux User Group)

“I have the second edition of *A Practical Guide to Linux® Commands, Editors, and Shell Programming* and am a big fan. I used it while working as a Cisco support engineer. I plan to get the third edition as soon as it is released. We will be doing a ton of command-line work on literally 1000 boxes (IMS core nodes). I feel you have already given me a lot of tools with the second edition. I want to get your new book as soon as possible. The way you write works very well for my style of learning.”

—Robert Lingenfelter
Support Engineer, VoIP/IMS

PRAISE FOR OTHER BOOKS BY MARK G. SOBELL

“Since I’m in an educational environment, I found the content of Sobell’s book to be right on target and very helpful for anyone managing Linux in the enterprise. His style of writing is very clear. He builds up to the chapter exercises, which I find to be relevant to real-world scenarios a user or admin would encounter. An IT/IS student would find this book a valuable complement to their education. The vast amount of information is extremely well balanced and Sobell manages to present the content without complicated asides and meandering prose. This is a ‘must have’ for anyone managing Linux systems in a networked environment or anyone running a Linux server. I would also highly recommend it to an experienced computer user who is moving to the Linux platform.”

—*Mary Norbury*
IT Director
Barbara Davis Center
University of Colorado at Denver
from a review posted on slashdot.org

“I had the chance to use your UNIX books when I when was in college years ago at Cal Poly, San Luis Obispo, CA. I have to say that your books are among the best! They’re quality books that teach the theoretical aspects and applications of the operating system.”

—*Benton Chan*
IS Engineer

“The book has more than lived up to my expectations from the many reviews I read, even though it targets FC2. I have found something very rare with your book: It doesn’t read like the standard technical text, it reads more like a story. It’s a pleasure to read and hard to put down. Did I say that?! :-)”

—*David Hopkins*
Business Process Architect

“Thanks for your work and for the book you wrote. There are really few books that can help people to become more efficient administrators of different workstations. We hope (in Russia) that you will continue bringing us a new level of understanding of Linux/UNIX systems.”

—*Anton Petukhov*

“Mark Sobell has written a book as approachable as it is authoritative.”

—*Jeffrey Bianchine*
Advocate, Author, Journalist

“Excellent reference book, well suited for the sysadmin of a Linux cluster, or the owner of a PC contemplating installing a recent stable Linux. Don’t be put off by the daunting heft of the book. Sobell has striven to be as inclusive as possible, in trying to anticipate your system administration needs.”

—*Wes Boudville*
Inventor

“*A Practical Guide to Red Hat® Linux®* is a brilliant book. Thank you Mark Sobell.”

—*C. Pozrikidis*
University of California at San Diego

“This book presents the best overview of the Linux operating system that I have found. . . . [It] should be very helpful and understandable no matter what the reader’s background: traditional UNIX user, new Linux devotee, or even Windows user. Each topic is presented in a clear, complete fashion and very few assumptions are made about what the reader knows. . . . The book is extremely useful as a reference, as it contains a 70-page glossary of terms and is very well indexed. It is organized in such a way that the reader can focus on simple tasks without having to wade through more advanced topics until they are ready.”

—*Cam Marshall*
Marshall Information Service LLC
Member of Front Range UNIX
Users Group [FRUUG]
Boulder, Colorado

“Conclusively, this is THE book to get if you are a new Linux user and you just got into RH/Fedora world. There’s no other book that discusses so many different topics and in such depth.”

—*Eugenia Loli-Queru*
Editor in Chief
OSNews.com

“I currently own one of your books, *A Practical Guide to Linux*[®]. I believe this book is one of the most comprehensive and, as the title says, practical guides to Linux I have ever read. I consider myself a novice and I come back to this book over and over again.”

—*Albert J. Nguyen*

“Thank you for writing a book to help me get away from Windows XP and to never touch Windows Vista. The book is great; I am learning a lot of new concepts and commands. Linux is definitely getting easier to use.”

—*James Moritz*

“I am so impressed by how Mark Sobell can approach a complex topic in such an understandable manner. His command examples are especially useful in providing a novice (or even an advanced) administrator with a cookbook on how to accomplish real-world tasks on Linux. He is truly an inspired technical writer!”

—*George Vish II*
Senior Education Consultant
Hewlett-Packard Company

“Overall, I think it’s a great, comprehensive Ubuntu book that’ll be a valuable resource for people of all technical levels.”

—*John Dong*
Ubuntu Forum Council Member
Backports Team Leader

“The JumpStart sections really offer a quick way to get things up and running, allowing you to dig into the details of the book later.”

—*Scott Mann*
Aztek Networks

“I would so love to be able to use this book to teach a class about not just Ubuntu or Linux but about computers in general. It is thorough and well written with good illustrations that explain important concepts for computer usage.”

—*Nathan Eckenrode*
New York Local Community Team

“Ubuntu is gaining popularity at the rate alcohol did during Prohibition, and it’s great to see a well-known author write a book on the latest and greatest version. Not only does it contain Ubuntu-specific information, but it also touches on general computer-related topics, which will help the average computer user to better understand what’s going on in the background. Great work, Mark!”

—*Daniel R. Arfsten*
Pro/ENGINEER Drafter/Designer

“I read a lot of Linux technical information every day, but I’m rarely impressed by tech books. I usually prefer online information sources instead. Mark Sobell’s books are a notable exception. They’re clearly written, technically accurate, comprehensive, and actually enjoyable to read.”

—*Matthew Miller*
Senior Systems Analyst/Administrator
BU Linux Project
Boston University Office
of Information Technology

“This is well-written, clear, comprehensive information for the Linux user of any type, whether trying Ubuntu on for the first time and wanting to know a little about it, or using the book as a very good reference when doing something more complicated like setting up a server. This book’s value goes well beyond its purchase price and it’ll make a great addition to the Linux section of your bookshelf.”

—*Linc Fessenden*
Host of The LinuxLink TechShow
tllts.org

“The author has done a very good job at clarifying such a detail-oriented operating system. I have extensive Unix and Windows experience and this text does an excellent job at bridging the gaps between Linux, Windows, and Unix. I highly recommend this book to both ‘newbs’ and experienced users. Great job!”

—*Mark Polczynski*
Information Technology Consultant

“Your text, *A Practical Guide to Ubuntu Linux*®, *Third Edition*, is a well constructed, informative, superbly written text. You deserve an award for outstanding talent; unfortunately my name is not Pulitzer.”

—*Harrison Donnelly*
Physician

“When I first started working with Linux just a short ten years or so ago, it was a little more difficult than now to get going. . . . Now, someone new to the community has a vast array of resources available on the web, or if they are inclined to begin with Ubuntu, they can literally find almost every single thing they will need in the single volume of Mark Sobell’s *A Practical Guide to Ubuntu Linux*®.

“I’m sure this sounds a bit like hyperbole. Everything a person would need to know? Obviously not everything, but this book, weighing in at just under 1200 pages, covers so much so thoroughly that there won’t be much left out. From install to admin, networking, security, shell scripting, package management, and a host of other topics, it is all there. GUI and command-line tools are covered. There is not really any wasted space or fluff, just a huge amount of information. There are screen shots when appropriate but they do not take up an inordinate amount of space. This book is information-dense.”

—JR Peck
Editor
GeekBook.org

“I have been wanting to make the jump to Linux but did not have the guts to do so—until I saw your familiarly titled *A Practical Guide to Red Hat*® *Linux*® at the bookstore. I picked up a copy and am eagerly looking forward to regaining my freedom.”

—Carmine Stoffo
Machine and Process Designer
to pharmaceutical industry

“I am currently reading *A Practical Guide to Red Hat*® *Linux*® and am finally understanding the true power of the command line. I am new to Linux and your book is a treasure.”

—Juan Gonzalez

“Overall, *A Practical Guide to Ubuntu Linux*® by Mark G. Sobell provides all of the information a beginner to intermediate user of Linux would need to be productive. The inclusion of the Live DVD of the Gutsy Gibbon release of Ubuntu makes it easy for the user to test-drive Linux without affecting his installed OS. I have no doubts that you will consider this book money well spent.”

—Ray Lodato
Slashdot contributor
www.slashdot.org

This page intentionally left blank

A PRACTICAL GUIDE TO LINUX® COMMANDS, EDITORS, AND SHELL PROGRAMMING

FOURTH EDITION

MARK G. SOBELL

COAUTHORED BY **MATTHEW HELMKE**

◆ Addison-Wesley

Boston • Columbus • Indianapolis • New York • San Francisco • Amsterdam • Cape Town
Dubai • London • Madrid • Milan • Munich • Paris • Montreal • Toronto • Delhi • Mexico City
São Paulo • Sydney • Hong Kong • Seoul • Singapore • Taipei • Tokyo

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The author and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact intlcs@pearson.com.

Visit us on the Web: informit.com/aw

Library of Congress Control Number: 201795249

Copyright © 2018 Mark G. Sobell

All rights reserved. Printed in the United States of America. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, request forms and the appropriate contacts within the Pearson Education Global Rights & Permissions Department, please visit www.pearsoned.com/permissions/.

ISBN-13: 978-0-13-477460-2

ISBN-10: 0-13-477460-4

*For Sandra,
Sage, Sedona, Philip, and Evan.
Love you tons!*

—Matthew Helmke

This page intentionally left blank

BRIEF CONTENTS

CONTENTS xv

PREFACE xxxv

1 WELCOME TO LINUX AND MACOS 1

PART I THE LINUX AND MACOS OPERATING SYSTEMS 21

2 GETTING STARTED 23

3 THE UTILITIES 49

4 THE FILESYSTEM 83

5 THE SHELL 127

PART II THE EDITORS 163

6 THE `vim` EDITOR 165

7 THE `emacs` EDITOR 221

PART III THE SHELLS 283

8 THE BOURNE AGAIN SHELL (`bash`) 285

9 THE TC SHELL (`tcsh`) 379

PART IV PROGRAMMING TOOLS 427

10 PROGRAMMING THE BOURNE AGAIN SHELL (`bash`) 429

11 THE PERL SCRIPTING LANGUAGE 529

- 12 THE PYTHON PROGRAMMING LANGUAGE 577
- 13 THE MARIADB SQL DATABASE MANAGEMENT SYSTEM 609
- 14 THE AWK PATTERN PROCESSING LANGUAGE 635
- 15 THE sed EDITOR 669

PART V SECURE NETWORK UTILITIES 687

- 16 THE rsync SECURE COPY UTILITY 689
- 17 THE OPENSsh SECURE COMMUNICATION UTILITIES 703

PART VI COMMAND REFERENCE 729

- COMMAND REFERENCE 731

PART VII APPENDICES 1035

- A REGULAR EXPRESSIONS 1037
- B HELP 1047
- C KEEPING THE SYSTEM UP-TO-DATE 1053
- D MACOS NOTES 1067

GLOSSARY 1081

FILE TREE INDEX 1135

UTILITY INDEX 1137

MAIN INDEX 1141

CONTENTS

PREFACE XXXV

CHAPTER 1: WELCOME TO LINUX AND MACOS 1

The History of UNIX and GNU–Linux	3
The Heritage of Linux: UNIX	3
Fade to 1983	3
Next Scene, 1991	4
The Code Is Free	5
Have Fun!	6
What Is So Good About Linux?	6
Why Linux Is Popular with Hardware Companies and Developers	9
Linux Is Portable	10
The C Programming Language	10
Overview of Linux	11
Linux Has a Kernel Programming Interface	12
Linux Can Support Many Users	12
Linux Can Run Many Tasks	12
Linux Provides a Secure Hierarchical Filesystem	12
The Shell: Command Interpreter and Programming Language	14
A Large Collection of Useful Utilities	15
Interprocess Communication	16
System Administration	16
Additional Features of Linux	16
GUIs: Graphical User Interfaces	16
(Inter)Networking Utilities	17
Software Development	18
Chapter Summary	18
Exercises	18

PART I THE LINUX AND MACOS OPERATING SYSTEMS 21

CHAPTER 2: GETTING STARTED 23

- Conventions Used in This Book 24
- Logging In from a Terminal (Emulator) 26
- Working from the Command Line 28
 - Which Shell Are You Running? 29
 - Correcting Mistakes 29
 - Repeating/Editing Command Lines 31
- su/sudo: Curbing Your Power (**root** Privileges) 32
- Where to Find Documentation 33
 - man: Displays the System Manual 33
 - apropos: Searches for a Keyword 35
 - info: Displays Information About Utilities 36
 - The **—help** Option 38
 - The **bash help** Command 39
 - Getting Help 39
- More About Logging In and Passwords 42
 - What to Do If You Cannot Log In 43
 - Logging In Remotely: Terminal Emulators, **ssh**, and Dial-Up Connections 43
 - Using Virtual Consoles 44
 - Logging Out 44
 - Changing Your Password 44
- Chapter Summary 46
- Exercises 47
- Advanced Exercises 48

CHAPTER 3: THE UTILITIES 49

- Special Characters 50
- Basic Utilities 51
 - ls**: Lists the Names of Files 52
 - cat**: Displays a Text File 52
 - rm**: Deletes a File 52
 - less Is more**: Display a Text File One Screen at a Time 53
 - hostname**: Displays the System Name 53
- Working with Files 53
 - cp**: Copies a File 53
 - mv**: Changes the Name of a File 54
 - lpr**: Prints a File 55
 - grep**: Searches for a String 56

head: Displays the Beginning of a File	57
tail: Displays the End of a File	57
sort: Displays a File in Order	58
uniq: Removes Duplicate Lines from a File	58
diff: Compares Two Files	59
file: Identifies the Contents of a File	60
l (Pipeline): Communicates Between Processes	60
Four More Utilities	61
echo: Displays Text	61
date: Displays the Time and Date	62
script: Records a Shell Session	62
unix2dos: Converts Linux Files to Windows and macOS Format	63
Compressing and Archiving Files	64
bzip2: Compresses a File	64
bzip2 and bunzip2: Decompress a File	65
gzip: Compresses a File	66
tar: Packs and Unpacks Archives	66
Locating Utilities	69
which and whereis: Locate a Utility	69
locate: Searches for a File	70
Displaying User and System Information	71
who: Lists Users on the System	71
finger: Lists Users on the System	72
uptime: Displays System Load and Duration Information	73
w: Lists Users on the System	73
free: Displays Memory Usage Information	74
Communicating with Other Users	75
write: Sends a Message	75
mesg: Denies or Accepts Messages	76
Email	77
Chapter Summary	77
Exercises	80
Advanced Exercises	81

CHAPTER 4: THE FILESYSTEM 83

The Hierarchical Filesystem	84
Directory Files and Ordinary Files	85
Filenames	86
The Working Directory	89
Your Home Directory	89
Pathnames	90
Absolute Pathnames	90
Relative Pathnames	91

Working with Directories	92
mkdir: Creates a Directory	93
cd: Changes to Another Working Directory	94
rmdir: Deletes a Directory	96
Using Pathnames	96
mv, cp: Move or Copy Files	97
mv: Moves a Directory	98
Important Standard Directories and Files	98
Access Permissions	100
ls -l: Displays Permissions	100
chmod: Changes Access Permissions	102
Setuid and Setgid Permissions	104
Directory Access Permissions	105
ACLs: Access Control Lists	106
Enabling ACLs	107
Working with Access Rules	108
Setting Default Rules for a Directory	110
Links	112
Hard Links	113
Symbolic Links	115
rm: Removes a Link	118
Dereferencing Symbolic Links	118
Chapter Summary	122
Exercises	124
Advanced Exercises	126
CHAPTER 5: THE SHELL	127
Special Characters	128
Ordinary Files and Directory Files	129
The Working Directory	129
Your Home Directory	129
The Command Line	130
A Simple Command	130
Syntax	130
Simple Commands	133
Processing the Command Line	134
Executing a Command	136
Editing the Command Line	137
Standard Input and Standard Output	137
The Screen as a File	138
The Keyboard and Screen as Standard Input and Standard Output	138
Redirection	140
Pipelines	145
Lists	149
Running a Command in the Background	150

Filename Generation/Pathname Expansion	152
The ? Special Character	153
The * Special Character	154
The [] Special Characters	155
Builtins	157
Chapter Summary	158
Utilities and Builtins Introduced in This Chapter	159
Exercises	159
Advanced Exercises	160

PART II THE EDITORS 163

CHAPTER 6: THE vim EDITOR 165

History	166
Tutorial: Using vim to Create and Edit a File	167
Starting vim	167
Command and Input Modes	169
Entering Text	170
Getting Help	171
Ending the Editing Session	174
The compatible Parameter	174
Introduction to vim Features	175
Online Help	175
Terminology	175
Modes of Operation	175
The Display	176
Correcting Text as You Insert It	177
Work Buffer	177
Line Length and File Size	177
Windows	178
File Locks	178
Abnormal Termination of an Editing Session	179
Recovering Text After a Crash	179
Command Mode: Moving the Cursor	181
Moving the Cursor by Characters	182
Moving the Cursor to a Specific Character	182
Moving the Cursor by Words	182
Moving the Cursor by Lines	183
Moving the Cursor by Sentences and Paragraphs	183
Moving the Cursor Within the Screen	184
Viewing Different Parts of the Work Buffer	184
Input Mode	185
Inserting Text	185

Appending Text	185
Opening a Line for Text	185
Replacing Text	185
Quoting Special Characters in Input Mode	186
Command Mode: Deleting and Changing Text	186
Undoing Changes	186
Deleting Characters	186
Deleting Text	187
Changing Text	188
Replacing Text	189
Changing Case	190
Searching and Substituting	190
Searching for a Character	190
Searching for a String	190
Substituting One String for Another	193
Miscellaneous Commands	197
Join	197
Status	197
. (Period)	197
Copying, Moving, and Deleting Text	197
The General-Purpose Buffer	198
Named Buffers	199
Numbered Buffers	199
Reading and Writing Files	200
Reading Files	200
Writing Files	200
Identifying the Current File	201
Setting Parameters	201
Setting Parameters from Within vim	201
Setting Parameters in a Startup File	202
The .vimrc Startup File	202
Parameters	202
Advanced Editing Techniques	206
Using Markers	206
Editing Other Files	207
Macros and Shortcuts	207
Executing Shell Commands from Within vim	208
Units of Measure	210
Character	210
Word	210
Blank-Delimited Word	211
Line	211
Sentence	211
Paragraph	212
Screen (Window)	213
Repeat Factor	213

Chapter Summary	213
Exercises	218
Advanced Exercises	219
CHAPTER 7: THE emacs EDITOR	221
History	222
Evolution	222
emacs Versus vim	223
Tutorial: Getting Started with emacs	224
Starting emacs	224
Exiting	226
Inserting Text	226
Deleting Characters	227
Moving the Cursor	228
Editing at the Cursor Position	230
Saving and Retrieving the Buffer	230
Basic Editing Commands	231
Keys: Notation and Use	231
Key Sequences and Commands	233
META-x : Running a Command Without a Key Binding	233
Numeric Arguments	233
Point and the Cursor	234
Scrolling Through a Buffer	234
Erasing Text	234
Searching for Text	235
Using the Menubar from the Keyboard	237
Online Help	238
Advanced Editing	240
Undoing Changes	240
Point, Mark, and Region	242
Cut and Paste: Yanking Killed Text	243
Inserting Special Characters	245
Global Buffer Commands	245
Visiting and Saving Files	247
Buffers	251
Windows	252
Foreground Shell Commands	254
Background Shell Commands	255
Major Modes: Language-Sensitive Editing	255
Selecting a Major Mode	256
Human-Language Modes	256
C Mode	259
Customizing Indention	262
Comments	263
Special-Purpose Modes	264

Customizing emacs	265
The .emacs Startup File	266
Remapping Keys	267
A Sample .emacs File	269
More Information	270
Access to emacs	270
Chapter Summary	270
Exercises	279
Advanced Exercises	280

PART III THE SHELLS 283

CHAPTER 8: THE BOURNE AGAIN SHELL (bash) 285

Background	286
Startup Files	288
Login Shells	288
Interactive Nonlogin Shells	289
Noninteractive Shells	289
Setting Up Startup Files	289
. (Dot) or source: Runs a Startup File in the Current Shell	290
Commands That Are Symbols	291
Redirecting Standard Error	292
Writing and Executing a Simple Shell Script	294
chmod: Makes a File Executable	295
#! Specifies a Shell	297
# Begins a Comment	298
Executing a Shell Script	298
Control Operators: Separate and Group Commands	299
; and NEWLINE Separate Commands	300
and & Separate Commands and Do Something Else	300
&& and Boolean Control Operators	301
() Groups Commands	302
\ Continues a Command	303
Job Control	304
jobs: Lists Jobs	305
fg: Brings a Job to the Foreground	305
Suspending a Job	306
bg: Sends a Job to the Background	306
Manipulating the Directory Stack	307
dirs: Displays the Stack	307
pushd: Pushes a Directory on the Stack	308
popd: Pops a Directory Off the Stack	309

Parameters and Variables	310
User-Created Variables	312
Variable Attributes	315
Keyword Variables	317
Special Characters	325
Locale	326
Time	330
Processes	333
Process Structure	333
Process Identification	334
Executing a Command	335
History	336
Variables That Control History	336
Reexecuting and Editing Commands	338
The Readline Library	345
Aliases	352
Single Versus Double Quotation Marks in Aliases	353
Examples of Aliases	354
Functions	356
Controlling bash: Features and Options	359
bash Command-Line Options	359
Shell Features	360
Processing the Command Line	364
History Expansion	364
Alias Substitution	364
Parsing and Scanning the Command Line	364
Command-Line Expansion	364
Chapter Summary	374
Exercises	376
Advanced Exercises	378
CHAPTER 9: THE TC SHELL (tcsh)	379
Shell Scripts	380
Entering and Leaving the TC Shell	381
Startup Files	382
Features Common to the Bourne Again and TC Shells	383
Command-Line Expansion (Substitution)	384
Job Control	388
Filename Substitution	388
Manipulating the Directory Stack	389
Command Substitution	389
Redirecting Standard Error	389
Working with the Command Line	390
Word Completion	391

- Editing the Command Line 393
- Correcting Spelling 394
- Variables 396
 - Variable Substitution 396
 - String Variables 396
 - Arrays of String Variables 397
 - Numeric Variables 398
 - Braces 401
 - Special Variable Forms 401
 - tcsh Variables 402
- Control Structures 408
 - if 409
 - goto 411
 - Interrupt Handling 412
 - if...then...else 413
 - foreach 414
 - while 416
 - break and continue 416
 - switch 417
- Builtins 418
- Chapter Summary 422
- Exercises 423
- Advanced Exercises 425

PART IV PROGRAMMING TOOLS 427

CHAPTER 10: PROGRAMMING THE BOURNE AGAIN SHELL (bash) 429

- Control Structures 430
 - if...then 431
 - if...then...else 435
 - if...then...elif 436
 - for...in 443
 - for 445
 - while 447
 - until 451
 - break and continue 453
 - case 454
 - select 460
 - Here Document 462
- File Descriptors 464
 - Opening a File Descriptor 465
 - Duplicating a File Descriptor 465

File Descriptor Examples	465
Determining Whether a File Descriptor Is Associated with the Terminal	468
Parameters	470
Positional Parameters	470
Special Parameters	475
Variables	479
Shell Variables	479
Environment, Environment Variables, and Inheritance	480
Expanding Null and Unset Variables	485
Array Variables	486
Variables in Functions	488
Builtin Commands	489
type: Displays Information About a Command	489
read: Accepts User Input	489
exec: Executes a Command or Redirects File Descriptors	493
trap: Catches a Signal	496
kill: Aborts a Process	499
eval: Scans, Evaluates, and Executes a Command Line	500
getopts: Parses Options	501
A Partial List of Builtins	503
Expressions	505
Arithmetic Evaluation	505
Logical Evaluation (Conditional Expressions)	506
String Pattern Matching	507
Arithmetic Operators	508
Implicit Command-Line Continuation	512
Shell Programs	513
A Recursive Shell Script	514
The quiz Shell Script	517
Chapter Summary	523
Exercises	525
Advanced Exercises	527

CHAPTER 11: THE PERL SCRIPTING LANGUAGE 529

Introduction to Perl	530
More Information	531
Help	531
perldoc	531
Terminology	533
Running a Perl Program	534
Syntax	536
Variables	538
Scalar Variables	540
Array Variables	541
Hash Variables	544

Control Structures	545
if/unless	546
if...else	548
if...elsif...else	549
foreach/for	549
last and next	551
while/until	552
Working with Files	554
Sort	558
Subroutines	559
Regular Expressions	562
Syntax and the =~ Operator	562
CPAN Modules	568
Examples	570
Chapter Summary	574
Exercises	574
Advanced Exercises	575

CHAPTER 12: THE PYTHON PROGRAMMING LANGUAGE 577

Introduction	578
Invoking Python	578
More Information	580
Writing to Standard Output and Reading from Standard Input	581
Functions and Methods	581
Scalar Variables, Lists, and Dictionaries	582
Scalar Variables	582
Lists	583
Dictionaries	587
Control Structures	588
if	589
if...else	589
if...elif...else	590
while	591
for	591
Reading from and Writing to Files	593
File Input and Output	593
Exception Handling	594
Pickle	596
Regular Expressions	597
Defining a Function	598
Using Libraries	599
Standard Library	599
Nonstandard Libraries	599

SciPy and NumPy Libraries	600
Namespace	600
Importing a Module	601
Example of Importing a Function	602
Lambda Functions	603
List Comprehensions	604
Chapter Summary	605
Exercises	606
Advanced Exercises	606

CHAPTER 13: THE MARIADB SQL DATABASE MANAGEMENT SYSTEM 609

History	610
Notes	611
Syntax and Conventions	612
More Information	614
Installing a MariaDB Server and Client	614
Fedora/RHEL (Red Hat Enterprise Linux)	614
Debian/Ubuntu/Mint	615
openSUSE	615
macOS	615
Client Options	615
Setting Up MariaDB	616
Assigning a Password to the MariaDB User Named <code>root</code>	616
Removing Anonymous Users	617
Running the Secure Installation Script	617
<code>~/.my.cnf</code> : Configures a MariaDB Client	617
<code>~/.mysql_history</code> : Stores Your MariaDB History	618
Creating a Database	618
Adding a User	619
Examples	620
Logging In	620
Creating a Table	621
Adding Data	622
Retrieving Data	623
Backing Up a Database	625
Modifying Data	626
Creating a Second Table	627
Joins	628
Chapter Summary	633
Exercises	633

CHAPTER 14: THE AWK PATTERN PROCESSING LANGUAGE 635

- Syntax 636
- Arguments 636
- Options 637
- Notes 638
- Language Basics 638
 - Patterns 638
 - Actions 639
 - Comments 639
 - Variables 639
 - Functions 640
 - Arithmetic Operators 641
 - Associative Arrays 642
 - printf 642
 - Control Structures 643
- Examples 645
- Advanced **gawk** Programming 662
 - getline**: Controlling Input 662
 - Coprocess: Two-Way I/O 665
 - Getting Input from a Network 666
- Chapter Summary 667
- Exercises 668
- Advanced Exercises 668

CHAPTER 15: THE sed EDITOR 669

- Syntax 670
- Arguments 670
- Options 670
- Editor Basics 671
 - Addresses 671
 - Instructions 672
 - Control Structures 673
 - The Hold Space 674
- Examples 674
- Chapter Summary 685
- Exercises 685

PART V SECURE NETWORK UTILITIES 687

CHAPTER 16: THE `rsync` SECURE COPY UTILITY 689

- Syntax 690
- Arguments 690
- Options 691
 - Notes 692
 - More Information 693
- Examples 693
 - Using a Trailing Slash (/) on `source-file` 694
 - Removing Files 694
 - Copying Files to and from a Remote System 696
 - Mirroring a Directory 697
 - Making Backups 697
 - Restoring a File 700
- Chapter Summary 700
- Exercises 701

CHAPTER 17: THE `OPENSSH` SECURE COMMUNICATION UTILITIES 703

- Introduction to OpenSSH 704
 - Files 704
 - More Information 706
- Running the `ssh`, `scp`, and `sftp` OpenSSH Clients 706
 - Prerequisites 706
 - JumpStart I: Using `ssh` and `scp` to Connect to an OpenSSH Server 706
 - Configuring OpenSSH Clients 707
 - `ssh`: Logs in or Executes Commands on a Remote System 709
 - `scp`: Copies Files to and from a Remote System 713
 - `sftp`: A Secure FTP Client 715
 - `~/.ssh/config` and `/etc/ssh/ssh_config` Configuration Files 715
- Setting Up an OpenSSH Server (`sshd`) 717
 - Prerequisites 717
 - Note 717
 - JumpStart II: Starting an OpenSSH Server 717
 - Authorized Keys: Automatic Login 717
 - Randomart Image 719
 - `ssh-agent`: Holds Your Private Keys 720
 - Command-Line Options 721
 - `/etc/ssh/sshd_config` Configuration File 722
- Troubleshooting 724

Tunneling/Port Forwarding	724
Forwarding X11	725
Port Forwarding	726
Chapter Summary	727
Exercises	728
Advanced Exercises	728

PART VI COMMAND REFERENCE 729

Utilities That Display and Manipulate Files	731
Network Utilities	732
Utilities That Display and Alter Status	733
Utilities That Are Programming Tools	734
Miscellaneous Utilities	734
Standard Multiplicative Suffixes	735
Common Options	736
The sample Utility	736
sample	Brief description of what the utility does <small>macOS</small> 737
aspell	Checks a file for spelling errors 739
at	Executes commands at a specified time 743
busybox	Implements many standard utilities 747
bzip2	Compresses or decompresses files 750
cal	Displays a calendar 752
cat	Joins and displays files 753
cd	Changes to another working directory 755
chgrp	Changes the group associated with a file 757
chmod	Changes the access mode (permissions) of a file 759
chown	Changes the owner of a file and/or the group the file is associated with 764
cmp	Compares two files 766
comm	Compares sorted files 768
configure	Configures source code automatically 770
cp	Copies files 772
cpio	Creates an archive, restores files from an archive, or copies a directory hierarchy 776
crontab	Maintains crontab files 781
cut	Selects characters or fields from input lines 784
date	Displays or sets the system time and date 787
dd	Converts and copies a file 790

-
- df Displays disk space usage 793
 - diff Displays the differences between two text files 795
 - diskutil Checks, modifies, and repairs local volumes *macOS* 800
 - ditto Copies files and creates and unpacks archives *macOS* 803
 - dmesg Displays kernel messages 805
 - dscf Displays and manages Directory Service information *macOS* 806
 - du Displays information on disk usage by directory hierarchy and/or file 809
 - echo Displays a message 812
 - expand/unexpand
 - Converts TABs to SPACES and SPACES to TABs 814
 - expr Evaluates an expression 816
 - file Displays the classification of a file 820
 - find Finds files based on criteria 822
 - finger Displays information about users 828
 - fmt Formats text very simply 831
 - fsck Checks and repairs a filesystem 833
 - ftp Transfers files over a network 838
 - gawk Searches for and processes patterns in a file 845
 - gcc Compiles C and C++ programs 846
 - GetFileInfo Displays file attributes *macOS* 851
 - grep Searches for a pattern in files 853
 - gzip Compresses or decompresses files 858
 - head Displays the beginning of a file 861
 - join Joins lines from two files based on a common field 863
 - kill Terminates a process by PID 866
 - killall Terminates a process by name 868
 - launchctl Controls the launchd daemon *macOS* 870
 - less Displays text files, one screen at a time 873
 - ln Makes a link to a file 878
 - lpr Sends files to printers 881
 - ls Displays information about one or more files 884
 - make Keeps a set of programs current 892
 - man Displays documentation for utilities 898
 - mc Manages files in a textual environment (aka Midnight Commander) 902
 - mkdir Creates a directory 909
 - mkfs Creates a filesystem on a device 911
 - mv Renames or moves a file 914
 - nice Changes the priority of a command 916

- nl Numbers lines from a file 918
- nohup Runs a command that keeps running after you log out 920
- od Dumps the contents of a file 921
- open Opens files, directories, and URLs *macOS* 926
- otool Displays object, library, and executable files *macOS* 928
- paste Joins corresponding lines from files 930
- pax Creates an archive, restores files from an archive, or copies a directory hierarchy 932
- plutil Manipulates property list files *macOS* 938
- pr Paginates files for printing 940
- printf Formats string and numeric data 942
- ps Displays process status 946
- renice Changes the priority of a process 951
- rm Removes a file (deletes a link) 953
- rmdir Removes directories 955
- rsync Securely copies files and directory hierarchies over a network 956
- scp Securely copies one or more files to or from a remote system 957
- screen Manages several textual windows 958
- sed Edits a file noninteractively 964
- SetFile Sets file attributes *macOS* 965
- sleep Creates a process that sleeps for a specified interval 967
- sort Sorts and/or merges files 969
- split Divides a file into sections 978
- ssh Securely runs a program or opens a shell on a remote system 980
- sshfs/curlftpfs
 - Mounts a directory on an OpenSSH or FTP server as a local directory 981
- stat Displays information about files 984
- strings Displays strings of printable characters from files 986
- stty Displays or sets terminal parameters 987
- sysctl Displays and alters kernel variables at runtime 991
- tail Displays the last part (tail) of a file 992
- tar Stores or retrieves files to/from an archive file 995
- tee Copies standard input to standard output and one or more files 1000
- telnet Connects to a remote computer over a network 1001
- test Evaluates an expression 1005
- top Dynamically displays process status 1008
- touch Creates a file or changes a file's access and/or modification time 1012
- tr Replaces specified characters 1014
- tty Displays the terminal pathname 1017
- tune2fs Changes parameters on an ext2, ext3, or ext4 filesystem 1018
- umask Specifies the file-creation permissions mask 1021

- uniq Displays unique lines from a file 1023
 - w Displays information about local system users 1025
- wc Displays the number of lines, words, and bytes in one or more files 1027
- which Shows where in PATH a utility is located 1028
 - who Displays information about logged-in users 1030
- xargs Converts standard input to command lines 1032

PART VII APPENDIXES 1035

APPENDIX A: REGULAR EXPRESSIONS 1037

- Characters 1038
- Delimiters 1038
- Simple Strings 1038
- Special Characters 1038
 - Periods 1039
 - Brackets 1039
 - Asterisks 1040
 - Carets and Dollar Signs 1040
 - Quoting Special Characters 1041
- Rules 1041
 - Longest Match Possible 1041
 - Empty Regular Expressions 1042
- Bracketing Expressions 1042
- The Replacement String 1042
 - Ampersand 1043
 - Quoted Digit 1043
- Extended Regular Expressions 1043
- Appendix Summary 1045

APPENDIX B: HELP 1047

- Solving a Problem 1048
- Finding Linux and macOS Related Information 1049
 - Mailing Lists 1049
- Specifying a Terminal 1050

APPENDIX C: KEEPING THE SYSTEM UP-TO-DATE 1053

- Using dnf 1054
 - Using dnf to Install, Remove, and Update Packages 1054
 - Other dnf Commands 1056

- dnf Groups 1057
- Downloading rpm Package Files Using dnf download 1058
- Configuring dnf 1058
- Using apt-get 1060
 - Using apt-get to Install, Remove, and Update Packages 1061
 - Using apt-get to Upgrade the System 1062
 - Other apt-get Commands 1062
 - Using apt Commands 1063
 - Repositories 1063
 - sources.list: Specifies Repositories for apt-get to Search 1063
- BitTorrent 1064

APPENDIX D: MACOS NOTES 1067

- Open Directory 1068
- Filesystems 1069
 - Nondisk Filesystems 1069
 - Case Sensitivity 1070
 - /Volumes 1070
- Extended Attributes 1070
 - File Forks 1071
 - File Attributes 1072
 - ACLs 1074
- Activating the Terminal META Key 1076
- Startup Files 1076
- Remote Logins 1076
- Many Utilities Do Not Respect Apple Human Interface Guidelines 1076
- Installing Xcode and MacPorts 1077
- macOS Implementation of Linux Features 1078

GLOSSARY 1081

FILE TREE INDEX 1135

UTILITY INDEX 1137

MAIN INDEX 1141

PREFACE

- Linux *A Practical Guide to Linux® Commands, Editors, and Shell Programming, Fourth 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.
- macOS This book also explains how to work with the UNIX/Linux foundation of macOS. 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 macOS. Where this book refers to Linux, it implicitly refers to macOS 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 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, Solus, 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.

Overlap If you read one of Mark Sobell's other books, *A Practical Guide to Fedora™ and Red Hat® Enterprise Linux*, or *A Practical Guide to Ubuntu Linux®*, or Matthew Helmke's *Ubuntu Unleashed* or *The Official Ubuntu Book*, you will notice some overlap between those books and the one you are reading now. The books cover similar information, presented from different perspectives and at different levels of depth depending on the intended audience for each book.

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 macOS
- **Power users** who want to explore the power of Linux or macOS from the command line
- **Professionals** who use Linux or macOS 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 macOS
- **UNIX users** who want to adapt their UNIX skills to the Linux or macOS environment
- **System administrators** who need a deeper understanding of Linux or macOS 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 macOS operating system
- **Programmers** who need to understand the Linux or macOS programming environment
- **Technical executives** who want to get a grounding in Linux or macOS

Benefits *A Practical Guide to Linux® Commands, Editors, and Shell Programming, Fourth Edition*, gives you a broad understanding of how to use Linux and macOS 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/macOS, 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 macOS, 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 *macOS*

tip The UNIX operating system is the common ancestor of Linux and macOS. 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 macOS. To make the content more readable, this book uses the term *Linux* to refer to both *Linux* and *macOS*. 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, Fourth 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 macOS are covered in detail, including `diskutil`, `ditto`, `dscl`, `GetFileInfo`, `launchctl`, `otool`, `plutil`, and `SetFile`.
- Descriptions of macOS **extended attributes** include **file forks**, **file attributes**, **attribute flags**, and **Access Control Lists (ACLs)**.
- Appendix D, “macOS Notes,” lists some differences between macOS 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 macOS**
Presents background information on Linux and macOS. This chapter covers the **history of Linux**, profiles the macOS **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 MACOS 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 macOS 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 MariaDB SQL Database Management System**

Introduces the widely used MariaDB/MySQL relational database management system (RDBMS). This chapter covers

- ◆ Relational database **terminology**;
- ◆ **Installing** the MariaDB 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 macOS 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 demon-

strate 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 826), `ftp` (page 841), and `sort` (page 971) to see how extensive these sections are. Some utilities, such as Midnight Commander (`mc`; page 902) and `screen` (page 958), include extensive discussion sections and tutorials.

PART VII: APPENDICES

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
 - ◆ `dnf`—Downloads software from the Internet, keeping a system up-to-date and **resolving dependencies** as it goes.
 - ◆ `apt-get`—An alternative to `dnf` for keeping a system current.
 - ◆ **BitTorrent**—Good for distributing large amounts of data such as Linux installation CDs and DVDs.
- **Appendix D—macOS Notes**
A brief guide to macOS 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 macOS.
- **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 macOS-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.

Register your copy of *A Practical Guide to Linux® Commands, Editors, and Shell Programming, Fourth Edition*, at informit.com/register for convenient access to downloads, updates, and/or corrections as they become available (you must log in or create a new account). Enter the product ISBN (9780134774602) and click Submit. Once the process is complete, you will find any available bonus content under Registered Products. If you would like to be notified of exclusive offers on new editions and updates, please check the box to receive email from us.

THANKS

As this is my (Matthew's) first edition of this book, I would like to begin by thanking Mark Sobell for trusting me with his creation. You have gifted me an excellent foundation and I am truly grateful. Enjoy your well-deserved retirement! I also want to thank Debra Williams Cauley and Mark Taub for approaching both me and Mark Sobell when he decided it was time to hand the book to someone else. Your trust in me is appreciated and not taken lightly.

I take responsibility for any errors and omissions in this book. If you find one or just have a comment, let me know (matthew@matthewhelmke.com), and I will fix it in the next printing. I inherited a fabulous amount of well-vetted content, and I have tested what is here while updating the text for this edition, but it is possible I have not done so perfectly and am happy to receive your kind assistance and corrections where needed.

The rest of this section is from Mark's previous edition. I share his gratitude to and appreciation of all who are mentioned here, many of whom have also worked with me on this edition.

Matthew Helmke
North Liberty, Iowa

(From the Third Edition, 2013)

First and foremost, I want to thank Mark L. Taub, Editor-in-Chief of the IT Professional Group at Pearson, 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 Pearson 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 Pearson 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; 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 Hagenaaars; and Casper Dik.

A Practical Guide to Linux® Commands, Editors, and Shell Programming, Fourth 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 on Twitter at twitter.com/marksobell.

Mark G. Sobell
San Francisco, California

1

WELCOME TO LINUX AND MACOS

IN THIS CHAPTER

The History of UNIX and GNU–Linux.....	3
The Heritage of Linux: UNIX	3
What Is So Good About Linux?.....	6
Overview of Linux	11
Additional Features of Linux	16

OBJECTIVES

After reading this chapter you should be able to:

- ▶ Discuss the history of UNIX, Linux, and the GNU project
- ▶ Explain what is meant by “free software” and list characteristics of the GNU General Public License
- ▶ List characteristics of Linux and reasons the Linux operating system is so popular
- ▶ Discuss three benefits of virtual machines over single physical machines