Mac for Linux Geeks

Tony Steidler-Dennison

Apress®

Mac for Linux Geeks

ISBN-13 (pbk): 978-1-4302-1650-6

ISBN-13 (electronic): 978-1-4302-1651-3

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Lead Editors: Frank Pohlmann and Michelle Lowman
Technical Reviewer: Peter O'Gorman
Editorial Board: Clay Andres, Steve Anglin, Mark Beckner, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Michelle Lowman, Matthew Moodie, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh
Project Manager: Sofia Marchant
Copy Editor: Marilyn Smith
Associate Production Director: Kari Brooks-Copony
Production Editor: Liz Berry
Compositor: Dina Quan
Proofreader: Lisa Hamilton
Indexer: Broccoli Information Management
Artist: April Milne
Cover Designer: Kurt Krames
Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail `orders-ny@springer-sbm.com`, or visit `http://www.springeronline.com`.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail `info@apress.com`, or visit `http://www.apress.com`.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at `http://www.apress.com/info/bulksales`.

As always, for my girls, Laurie and Mia

Contents at a Glance

About the Author xiii
About the Technical Reviewer xv
Preface xvii

CHAPTER 1 The Backstory 1
CHAPTER 2 The Comparison: Linux vs. Mac OS X 17
CHAPTER 3 Dual-Booting and Virtualization 41
CHAPTER 4 Building Out the Linux Environment 65
CHAPTER 5 Using the Many Apple and Linux Tools 81
CHAPTER 6 Routine Mac OS X System Administration 125
CHAPTER 7 Backup, Security, and Automation 155
CHAPTER 8 Mac OS X and Code 195
CHAPTER 9 Hybridizing Your System 233

INDEX 259

Contents

About the Author xiii
About the Technical Reviewer xv
Preface xvii

CHAPTER 1 The Backstory 1

Of Macros and Manuals: UNIX 1
From Assembly to C 2
Macros and Pipes 2
User Manuals 3
The Fork: BSD 4
1BSD to 4BSD 4
Licensing Issues 4
The Enthusiast and the Marketer: Apple Computer 5
Homebrew Days 5
Apple I to Lisa 7
And Finally, the Mac 8
The Convergence: Mac OS X 9
NeXTStep 9
Back at Apple 10
Why BSD in Mac OS X? 11
History 11
Portability 12
Open Source Base 12
Economics 13
Extensibility 13
How Is BSD Implemented in Mac OS X? 14
Why Switch from Linux to Mac? 14
Hardware Control 15
Common Code 15
Release Stability 15
Summary 15

CHAPTER 2 The Comparison: Linux vs. Mac OS X 17

Mac OS X and Linux Filesystems 17
The Apple Filesystem 17
Filesystem Layouts 22
The ext2/ext3 Filesystem in Linux. 28
Comparison of HFS+ and ext2/ext3 29
Permissions in Mac OS X 30
File Permissions 30
Root and Administrative Access 31
Terminal Access in Mac OS X 32
Starting Bash 32
Setting Linux System Variables in Mac OS X 33
Interfaces in Mac OS X 34
Configuring Ethernet Interfaces from the Command Line. 34
Using the GUI to Configure Ethernet Interfaces. 35
Devices and Drives 38
Accessing Devices and Drives Through the GUI 38
Accessing Devices and Drives from the Command Line. 39
Summary. 39

CHAPTER 3 Dual-Booting and Virtualization 41

Dual-Booting Linux and Mac OS X 41
Loading Linux with rEFIt 42
Installing Linux Using Boot Camp 45
Partitioning from the Command Line 50
Removing a Linux Partition. 51
Virtual Linux 54
Using VMware. 54
Using VirtualBox 63
Summary. 63

CHAPTER 4 Building Out the Linux Environment 65

Xcode Tools 65
Xcode Installation. 66
An Overview of the Xcode Tool Set. 70
Online Linux Tools 71
MacPorts 71
Fink. 75
Summary. 80

CHAPTER 5 Using the Many Apple and Linux Tools 81

A Brief Overview of Graphics and Multimedia on the Mac 81
Core Graphics 82
Core Video 84
Quartz Composer 86
Built-in Mac OS X Multimedia Tools 86
iPhoto 86
iMovie and iDVD 89
iWeb 92
GarageBand 94
Third-Party Multimedia Tools 98
The Adobe Multimedia Tools 99
Mac OS X Third-Party Multimedia Summary 103
Open Source Multimedia Tools 104
Graphics Editing with GIMP 104
Audio Editing with Audacity 106
Open Source Multimedia Summary 108
Office and Productivity Tools in Mac OS X 108
Microsoft Office for Mac 109
The Mac iWork Tools 111
Open Source Productivity Tools 117
OpenOffice.org 117
NeoOffice 118
Summary 122

CHAPTER 6 Routine Mac OS X System Administration 125

Using the Shell 125
Changing the Default Shell 126
Using UNIX Administration Tools and Commands 128
System Monitoring 129
Using Activity Monitor 129
Viewing System Processes with top 132
Listing Processes with ps 134
User Maintenance 135
Managing User Accounts Using System Preferences 135
Managing Users Using the Command Line 138

Log Review and Maintenance 140
Log Location and Naming Conventions 140
Reviewing Log Files with the Console Application 140
Managing Tasks with launchd 143
Administering Shared Resources 145
Mac OS X and Web Servers 145
Printer Sharing 148
SMB File Sharing 151
NFS File Sharing 151
Summary 152

CHAPTER 7 Backup, Security, and Automation 155

Backup and Recovery Overview 155
The Mac Approach to Backup and Recovery 156
Time Machine Backups 156
Backups with Carbon Copy Cloner 162
SuperDuper for Simple Backups 166
Mozy and Other Off-Site Backup Options 170
The Linux Approach to Backup and Recovery 178
Using dd to Copy Data 179
Using rsync to Synchronize Files 180
Security 182
Configuring Security Through System Preferences 183
Using ipfw As a Firewall 186
Using WaterRoof: An ipfw Front End 188
Summary 192

CHAPTER 8 Mac OS X and Code 195

Using Xcode 195
Creating an Application with Xcode 197
Working in the Main Xcode Window 200
Debugging with Xcode 203
Xcode and Other Application Development Tools 207
Xcode and Java 208
Xcode and Python 209
Xcode and Ruby 211
Xcode and PHP 213

Scripting 216
Using AppleScript 216
Creating Scripts with the Script Editor 217
Using Other Scripting Languages 218
Code Maintenance and Revision Control 218
Introducing Subversion 219
Using Subversion from the Command Line 221
Using Subversion GUI Front Ends 222
Managing Changes with Git 230
Summary 232

CHAPTER 9 Hybridizing Your System 233

How BSD and Linux Differ 234
Distribution vs. Operating System 234
Runlevels and System Startup 234
Licensing 236
Kernel Customization and Compilation 237
Setting Up the Build Environment 237
Building the Kernel 240
Porting UNIX Apps to the Mac 241
Why Port? 241
Good Practice 243
Installing the Development Environment 244
Creating Makefiles 246
Installing Linux Desktop Environments on the Mac 249
Installing GNOME 249
Installing KDE 251
Summary 257

INDEX 259

About the Author

TONY STEIDLER-DENNISON is a longtime Linux user and a recent convert to the Mac. He has coauthored two books on Linux and has written frequently for *Linux Journal* and *LinuxWorld*. He is also the host and producer of The Roadhouse Podcast, a weekly hour of "the finest blues you've never heard," at `http://roadhousepodcast.com/`.

Tony is a systems engineer with Rockwell Collins, Inc., leveraging open source technologies in communication products for commercial aviation. He and his family make their home in Iowa City, Iowa.

About the Technical Reviewer

PETER O'GORMAN is a software engineer. He first used a Mac in 1988, when he was a student at the University of Limerick, Ireland. Although he still uses one today, he also uses Linux, Solaris, AIX, and HP-UX on a daily basis.

Peter has contributed to a number of open source projects, including Apple's Darwin, Fink, MacPorts, GNU libtool, GCC, and Autoconf. He continues to make contributions to these projects when time allows.

Originally from Ireland, Peter has traveled the world, living in the United Kingdom, Australia, Hong Kong, and Japan before moving to Winnipeg, Canada, where he shivers through winters with his wife and daughter.

Preface

I didn't come to the Mac overnight, though it must have seemed that way to my friends and family. One day, I was extolling the virtues of Linux and open source; the next, I was talking about the Macintosh platform with nearly as much vigor.

My first computer was an Atari 1040ST, a stunning piece of machinery for 1986. It was available for less than $1,500 and came equipped with an entire megabyte of RAM. I hadn't been in the fledgling computer club in school during the 70s, and I really couldn't put my finger on why I had any interest at all in computers. In fact, I'm not sure I had even seen one before taking a sales job at a Federated electronics store. Computers were for geeks, after all (or, as we so mockingly called them in school, nerds). But from the first time I connected to CompuServe, computing had its hooks in me deeply. In just a few short weeks, I had made friends with a fellow computer enthusiast in our city of Arlington, Texas, and we managed to battle it out in mock dogfights online with a crude flight simulator for hours on end. Our families quickly tired of the sound of the modem when they called. According to my wife, I had clearly developed a substance-abuse problem. I had. The substance was silicon.

My strong affinity for computing continued, but by 1996, I had begun to tire of the install/reboot/blue screen sequence of the young Windows 95. I acquired a Toshiba Infinia, a reasonably stout machine for its time, and often lugged a thick, heavy Compaq laptop to and from work. But I felt constrained, limited by the roadblocks that seemed built into Windows.

One day at work, I commiserated with a friend who happened to work in our company's IT department. He nodded his agreement without saying much, pulled on his ponytail, and let me finish. Then, almost casually, he mentioned, "I've been playing with this new operating system. It's called Linux. Been out for a few years. It's not easy to get configured, but it's pretty powerful and interesting."

Challenging, powerful, and interesting—that description caught my attention. "Where can I get it?" I asked.

"Net. It's free."

It took just a few all-nighters at home to research this new operating system, find and download the install diskettes (the Infinia had no CD burner, and ISO images of the few Linux distributions were few and far between), and fail miserably at the first several installation attempts. With each failed installation, I would give up and reinstall Windows, adding yet more hours to the already painful process. But with each attempt, the challenge rose a bit higher, until I resolved that no simple computer was going to defeat me. I researched, learned, and researched some more. When I discovered a HOWTO on dual-booting Windows and Linux, the lights started to come on. Shortly after, I got my first good installation of Red Hat 4.0, dual-booting with Windows 95, and was off to the races. I made a commitment when that installation was complete that I would use Windows only when absolutely necessary, and that it wouldn't be necessary too often. And I found my powerful, flexible, challenging operating system of choice. In short, Linux revived my love of computing, making my wife once again a victim of my renewed substance-abuse problem.

Within a few short years, I had left a legal-field programming position with a large Iowa insurance company to pursue dreams of dot-com dollars. My writing experience and abilities got me in the door of the first startup, a company that was founded by Anton Olsen, my Linux friend and mentor from the previous company. The shop was entirely open source, and I reveled in the atmosphere of a small company where, in one minute, I could draft and send out press releases, while the next brought yet another learning experience in a room full of open source gurus. The company was short-lived, but the experience infused me with even more passion for Linux, for programming, and for the unbridled idealism of the open source philosophy.

Less than a year later, I began writing a daily Linux e-mail newsletter for Chris Pirillo, Lockergnome's Penguin Shell. For my day job, I took a position building and configuring Linux-based computers—not just any computers, but computers to control observatory-grade robotic telescopes built by a company in my hometown of Iowa City. I also helped assemble those telescopes and flew around the world to install them at dark locations around the planet. When that small company failed, I became a partner in another, building and repairing computers, with a special interest and expertise in Linux. Over the next four years, I chased Linux through a revival of the telescope company, a presidential campaign, online shopping, online real estate, online document scanning, and finally, into the world of commercial aviation, where I still work today. It's not always been the best living; Linux has, nonetheless, been very good to me.

During those Linux-chasing years, I was aware of the other computing platforms outside the open source realm. In fact, as the necessity of home computers grew, snaring friends with a new desire to discover the Web, I often recommended Macintosh machines as their first. Although I had barely even seen a Mac, I knew they had a reputation for user-friendliness and some serious brand loyalty. In return for the recommendation, those friends planted a very small seed in my mind. I watched as they became real computer enthusiasts and hard-core advocates for the Macintosh platform. I saw in them a dedication that I understood. It wasn't much different from the one I felt for Linux. Although they didn't have (or require) the hard-core skills I had picked up over the years, there was no doubt that they were enjoying their computing experiences. That was the feeling that had drawn me into the Linux world. I enjoyed seeing it in others, even on the Mac.

The real seed for the move to the Mac came in late 2003, when I joined the presidential campaign of General Wesley Clark in Little Rock, Arkansas. I was the second hire in the tech staff and gladly worked from my Fedora-installed Dell Inspiron laptop. As we filled the tech department to what eventually totaled 18 staff members, more and more of them arrived in Little Rock with MacBooks under their arms. And those small computers just worked. I watched coworkers switch effortlessly between a stunning GUI and the command line—whatever suited their needs for the particular task at hand. All the Linux commands that I used so frequently were available, and the hardware and desktop were beautifully designed. Much of the technical heavy lifting in that campaign was done on Macs, including all the web design, large chunks of the database design, and significant portions of the PHP development. I left Little Rock in February 2004, knowing that, at some point in the future, I would own a Mac.

While it took a few years, I did fulfill that promise to myself in December 2006. I purchased a Mac mini, one of the 1.83 GHz Intel Core Duo variety, with 2GB of RAM. At the time, I was nearly two years into the production of The Roadhouse Podcast, a weekly hour of "the finest blues you've never heard." Although I understood that Mac OS X was solidly designed and built around the BSD operating system, I had some concerns about moving the production of

the podcast to the Mac. I had landed on a routine with the show that was working well, though large periods of time were spent waiting for my old 800 MHz P3 Linux box to churn through encoding and conversion tasks. I had landed on a set of open source tools that met all my needs for the show, both practical and esoteric, and had no desire or time to learn a new Mac tool set. In the first week with the Mac mini on my desktop, I downloaded and installed those tools, either from the Web or via the MacPorts utility. And, on that first Saturday, my production time was actually reduced by a full two hours. The open source tools worked equally well on the Mac, and the solid hardware took less than half the time to accomplish the most CPU-intensive processes involved in assembling the show. In short, I was hooked.

It was at that point that I began to evangelize with friends and fellow computing professionals about the power of the Mac OS X system. Those who knew me well understood that efficiency was always my primary goal. They knew that for many years, an acceptable level of efficiency and stability were possible only with Linux. And, while they may have scratched their heads at the suddenness of my conversion, that conversion really wasn't, as it appeared, a transient overnight revelation. It had been a long time coming and was capped by the BSD base of Mac OS X.

It was only after purchasing the Mac mini that I realized the two pieces of Macintosh history that made this easy transition possible. The first was the introduction of Mac OS X. It was the first version of the Macintosh operating system to fully utilize BSD at its core. While Apple has added much to BSD for its version of Mac OS X, the full functionality of the renowned UNIX operating system remains. The classic set of UNIX tools is readily accessible and is also fully extensible via the MacPorts and Fink utilities. These utilities are similar to the `apt` tool in Debian-based Linux distributions. The MacPorts repositories, in particular, continue to add new tools, both for the command line and the GUI desktop. It's possible to accomplish many tasks on the Mac with either "for-pay" tools created specifically for the Mac OS X platform or open source tools. And, with a known hardware profile, developers of either application type can focus on a single platform, removing most of the obstacles found in Windows development and eliminating the instability of unknown hardware and peripherals. (Many Mac users have made the case that Windows installations on Intel Macs are, in fact, the best Windows installations they have used.) In other words, Mac developers know what hardware will be used to run their applications. Unlike the ad hoc nature of Windows hardware, developers can make full use of the Mac hardware.

The other enabling event in the history of the Mac was the transition from Motorola to Intel processors in 2006. That transition brought to bear the full weight of the existing BSD codebase. It also unleashed the full power of BSD on the Macintosh platform. In combination, the powerful capabilities of BSD on a known and native hardware platform pushed Mac OS X and the Macintosh well into the mainstream for serious developers.

My transition to the Mac has been, for all intents and purposes, seamless. Much like the move from Windows to Linux in 1996, the change in platform has breathed new life into both my recreational and vocational computer experiences. I've come the closest yet to that long-time goal of complete computing efficiency. I didn't need to relearn tools I relied upon in my Linux work. While I did need small adjustments to the structure of Mac OS X, the core functionality of the tools was virtually the same as I had spent years learning and using in Linux. Nestled within the clean and friendly designs of both hardware and software, the common UNIX codebase of Mac OS X made it possible to move, overnight, from one platform to the other, and to enjoy an even higher level of efficiency.

If I've learned anything about the greater Linux community, it's that we are, as a group, extreme Tux loyalists. There's a sincere dedication on the part of many to the grandeur and idealism of the free and open source software (FOSS) philosophy as presented by Eric Raymond's seminal *The Cathedral and the Bazaar*. Longtime Linux users may find it difficult to make the mental shift from that idealism to an acceptance of a proprietary operating system—even one that relies so heavily on a FOSS core. In my own experience, I've been no less the loyal idealist.

But for many, there's a deeper issue at play. An evangelist's attitude regarding FOSS is only as good as the efficiency of the code itself. If FOSS applications are cranky, inefficient, and generally difficult to implement, those applications will never make it beyond the horizon of hard-core users. Despite a history approaching 15 years, for example, the Linux desktop has yet to find its way into the mainstream, where the underlying FOSS principles can reach full fruition. Making computing more affordable and accessible is a goal that, essentially, starts with the usability of the operating system and the user interface. In other words, the greater FOSS goals of spreading the power of computing without regard to economic circumstance are entirely reliant on making the entire computing platform—hardware and software—efficient and usable for all who choose it. If a computing system is so complex as to be accessible only to geeks, it's unlikely that those goals will ever be accomplished.

Almost without exception, I've found the Mac OS X experience to be rewarding. The tools work. The hardware is stable and robust. And, like the proverbial icing on the cake, the GUI is pretty, intuitive, and very functional. There's a reason Macs have gained their reputation in the world of multimedia. All those elements are critical in an operating system that will spend many, many hours churning out beautiful graphics, editing video, and making music. Not coincidentally, those uses tend to put a complete computing system to its full test. Processor-intensive applications shine on the Mac. That's a function of known hardware for which developers can write code with relative ease. Most certainly, that's one of many reasons why open source code runs so well and so easily on the Mac, too. It's also a function of a common set of development tools, included with every Mac OS X operating system disc. A well-executed operating system based on BSD, a known hardware platform, a view toward design and ease of use—these are the leading reasons for my personal migration to the Macintosh and Mac OS X, and the reasons I've chosen to present you with this book.

I'll say it right up front: Mac OS X just works. It has the power, the tools, and the stability to rival any operating system—Linux and UNIX included. If you're interested enough to have picked up this book, you're about to enter a new and thoroughly satisfying computer experience.

Mac for Linux Geeks is based on my own personal experience in the transition from Linux to the Mac. Those of us who have spent time in the Linux realm tend to view and use computers a bit differently than the rest of the world (as epitomized by the old joke: "What are the two best things to come out of Berkeley? UNIX and LSD."). Personally, I love the power and flexibility of the command line. Mac OS X has that. I love the ability to dash off a quick script to solve an immediate problem, and then finding that it works in other situations just as well. Mac OS X has that. I like to compile my own software with options to tailor it specifically for my use or my machine. Mac OS X has that. I'm almost cranky in my devotion to source control. Mac OS X has that. I want a nice visual representation of the hour-long audio files I knock out each week in The Roadhouse Podcast. Mac OS X has that. I want a filesystem layout that makes sense in light of my longtime Linux use. Mac OS X has that, too. I want complete control and flexibility in my operating system environment. Mac OS X certainly has that. In the pages

that follow, we'll walk through these personal requirements and many others for the millions of Linux users around the world. But be aware that if you've purchased this book, it's more likely than not that your days with our old pal Linux are numbered.