Beginning Unix®

Paul Love, Joe Merlino, Craig Zimmerman,
Jeremy C. Reed, and Paul Weinstein

WILEY
Wiley Publishing, Inc.

Beginning Unix®

Beginning Unix®

Paul Love, Joe Merlino, Craig Zimmerman,
Jeremy C. Reed, and Paul Weinstein

WILEY
Wiley Publishing, Inc.

Beginning Unix®

Published by
Wiley Publishing, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Published simultaneously in Canada

ISBN 13: 978-0-7645-7994-3

ISBN 10: 0-7645-7994-0

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

1MA/RR/QU/QV/IN

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Website is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Website may provide or recommendations it may make. Further, readers should be aware that Internet Websites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services or to obtain technical support, please contact our Customer Care Department within the U.S. at (800) 762-2974, outside the U.S. at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Library of Congress Cataloging-in-Publication Data: Available from publisher

About the Authors

Paul Love (Cincinnati, OH), CISSP, CISA, CISM, Security+, has been in the IT field for 15 years. Paul holds a Masters of Science degree in Network Security and a Bachelor's in Information Systems. He has co-authored two Linux security books, contributed to multiple Linux/Unix books, and has been the technical editor for over 10 best selling Linux and Unix books. Paul also ran a successful Linux portal site during the dot com era and has been an avid Unix/Linux user and administrator both professionally and as a hobby for many years.

Joe Merlino (Boston, MA) is an experienced system administrator with Unix and Linux for more than a decade.

Craig Zimmerman (New York, NY) manages UNIX, Macintosh, and Windows systems for Spontaneous, a post-production company in New York City. He previously worked at Chiat/Day helping build the world's most famous virtual advertising agency, managing and networking Unix and Macintosh systems in multiple offices.

Jeremy C. Reed (Marysville, WA) is a programmer, a member of NetBSD, and has actively taught FreeBSD, NetBSD, and OpenBSD administration classes for the past three years.

Paul Weinstein (Chicago, IL) has worked on various Unix-based computing platforms, from the mainframe (Harris HCX-9) to the desktop (Powerbook G4) and has developed applications on just about all of the current major branches of Unix in the course of the past 10 years. Recently he has been focusing a lot of his attention on developing and integrating Web-based systems using tools such as Linux, Apache, MySQL, and Perl, and in doing so has brought his unique understanding to a wide range of computing environments ranging from public elementary schools to pioneering open source companies. Currently, Paul works as President and Chief Consultant for the computer consulting firm Kepler Solutions, Inc.

David Mercer (Cape Town, South Africa) is a long-time Unix user and PHP programmer who contributed to *Beginning PHP4* and *Beginning PHP5*. He has maintained a keen interest in all things open source ever since he managed to put together a working Beowulf cluster by nicking old computer parts from colleagues and assembling them under his desk.

Credits

Acquisitions Editor
Debra Williams

Development Editor
Maryann Steinhart

Production Editor
Felicia Robinson

Technical Editors
Robert Berg
John Kennedy
David Mercer
David Bruce

Copy Editor
Publication Services

Editorial Manager
Mary Beth Wakefield

Vice President & Executive Group Publisher
Richard Swadley

Vice President and Publisher
Joseph B. Wikert

Project Coordinator
Erin Smith

Graphics and Production Specialists
April Farling
Carrie A. Foster
Lauren Goddard
Denny Hager
Lynsey Osborn

Quality Control Technicians
Amanda Briggs
Brian H. Walls

Proofreading and Indexing
TECHBOOKS Production Services

Contents

Acknowledgments

I would like to thank my family and those who mentored me throughout my career.

I would like to thank the staff at Wiley, particularly Debra Williams Cauley, who helped get this book started and whose participation during the writing of this book was instrumental in its completion. I would also like to thank Maryann and the technical editors whose tough first reviews and great insight helped develop the book into a far greater work. All others at the Wrox team who helped make this book a better product through their input or editing are greatly appreciated.

Finally, I would like to thank all the developers of the Unix systems and their derivatives. Their tireless pursuit of excellence has given us one of the most elegant and stable operating systems available today.

—Paul Love

Introduction

The new millennium has seen many changes in many areas of computing, from new forms of storage with massive amounts of storage space, to systems that are far more powerful than the first computer users could have ever imagined. Designed and initially created more than 30 years ago, the Unix operating system has been part of the evolution of computers, so it's no accident that Unix is still one of the most popular operating systems for mission-critical tasks.

Unix is the basis for some of the most-used operating systems today, from Apple's Mac OS X to Linux to the more commonly known Unix versions, such as Sun's Solaris Unix and IBM's AIX. Today many of the versions of Unix are available free to users and corporations, allowing for a larger use base than many had imagined when Unix was first being developed. Unix is now seen as a user-friendly, very secure, and robust operating system rather than the cold, command line–only operating system once thought to be useful only to computer experts.

Beginning Unix covers all basic aspects of the Unix operating system. What is unique about this book is that it covers not only the standard Unix systems, such as Sun's Solaris and IBM's AIX, but also Unix derivatives, such as Apple's Mac OS X and the various Linuxes. Additionally, this book includes a unique conversion section explaining how to convert Mac OS X–specific or Windows operating systems commands that you may already know into their Unix equivalents, making the transition from other operating systems much easier.

This book also includes a CD-ROM with the KNOPPIX operating system. This fully functional version of Linux enables you to restart your computer into a Linux environment. KNOPPIX requires no technical experience, and it will not damage or modify your current operating system. Using KNOPPIX is an easy way for you to follow along with the book, learning Unix without the consequences of having to lose any data or operating systems on your computer.

Who Is This Book For?

This book is for anyone who is interested in understanding the concepts and operation of the Unix operating system, including any of the Unix derivatives available today (Apple OS X, Linux, or BSD, for example). It is designed for absolute beginners to the Unix operating system, including those who have only worked with the many graphical user interfaces available for the different Unix systems (Apple's Aqua interface, KDE, GNOME, and so forth). This book can also be useful for veteran Unix users, because no one knows everything about Unix, as a refresher on known concepts or as a tool to fill gaps in some knowledge areas.

No assumptions are made about the reader's skill level or prior use of computers. If you have used computers and other operating systems such as Mac OS X or Microsoft Windows, you will understand some of the concepts faster, but all readers will gain some insight from this book, regardless of their present expertise.

What Does This Book Cover?

This book covers all versions of Unix in their most basic form, as well as commands and concepts common to all versions of Unix and its derivatives, including:

- ❑ Apple's Mac OS X
- ❑ Red Hat Linux
- ❑ Mandrakelinux
- ❑ IBM's AIX
- ❑ Any version of Linux
- ❑ Any version of BSD (FreeBSD, OpenBSD, NetBSD)

Special emphasis is placed on Sun's Solaris, Mac OS X, and Linux because they are the most popular available. The different versions of Unix utilize the same principles and commands with small differences, so any version of Unix can be used with this book.

This book also covers basic programming, including shell scripting and Perl programming, which enable you to automate your system as much as possible—one of the strengths of the Unix operating system. The coverage of these programming concepts creates a firm foundation for more advanced programming covered by other books.

How This Book Is Structured

This book presents basic concepts of the Unix operating system first, progressing to more advanced topics and programming later in the book. If you are familiar with the concepts or commands covered in one chapter, you can simply skip to one that has information you need to learn.

Chapters 1 through 4 provide the fundamental information you need to understand Unix methodology, how Unix is designed, and the basics of logging in to and out of a Unix system.

- ❑ **Chapter 1: Unix Fundamentals**. The basics of Unix, including the history and terminology as well as some of the core concepts of Unix design and philosophy. This chapter helps you understand some of the culture behind the Unix operating system.
- ❑ **Chapter 2: First Steps**. This chapter describes the very first steps you must take to utilize the Unix operating system effectively, including what occurs during the Unix boot process, how to log in, and how the user environment (shell) is structured, as well as how to shut down a Unix system properly.
- ❑ **Chapter 3: Understanding Users and Groups**. Learning how users and groups work within the system is crucial to understanding how you can effectively use your system. This chapter covers all aspects of user accounts and groups, including how to add, modify, and delete user accounts and how to become another user with the `su` command.
- ❑ **Chapter 4: File System Concepts**. The Unix file system is one of the most critical components of the Unix system as a whole. The file system allows you to store and manipulate your files. This

chapter shows you what the Unix file system is and how to use it from a user and system administrator point of view. You will learn how to utilize the file system effectively, so that you can prevent some of the common problems associated with file system management.

Chapters 5–7 put you to work, from customizing your working environment to editing files on Unix. These chapters extend your repertoire of Unix commands.

- ❑ **Chapter 5: Customize Your Working Environment**. The shell is the primary environment that you use for day-to-day work in Unix. Unix offers a multitude of ways to customize your working environment to suit your needs and whims. This chapter goes over the many different configuration options available for users in many of the different Unix shells.
- ❑ **Chapter 6: Unix Commands In-Depth**. Unix has hundreds of different commands that do many tasks. This chapter provides a foundation for some of the most commonly used commands you will need to understand in order to use the system effectively for day-to-day work.
- ❑ **Chapter 7: Editing Files with Vi**. The vi editor is one of the oldest and most widely used text editors in Unix. It is commonly seen as a monolithic and difficult-to-use editor, but as you will learn, it is a very powerful and fast way to edit files. This chapter explores all aspects of using the vi editor to create and edit files effectively.

With a good foundation in place, you're ready to move on to more-advanced topics. Chapters 8–11 discuss how to use some powerful Unix tools, how to manage processes, and how to schedule programs to run at specific times. Chapter 12 takes on the important subject of security.

- ❑ **Chapter 8: Advanced Tools**. This chapter introduces the concept of regular expressions and covers some of the more advanced tools available to the Unix user.
- ❑ **Chapter 9: Advanced Unix Commands: Sed and AWK**. `sed` and `awk` are two very powerful tools that enable a user to manipulate files in an efficient manner. These commands are essential, and you will find yourself using them frequently. This chapter goes from the ground up in showing you how to use these commands.
- ❑ **Chapter 10: Job Control and Process Management**. This chapter covers the basics of Unix processes and how to control and manage these crucial components of the Unix operating system. As an extension of processes, job control is reviewed and explained.
- ❑ **Chapter 11: Running Programs at Specified Times**. Running programs at specified times without user or administrator intervention provides a user or administrator with the capability to run programs with minimal system impact when the fewest users are utilizing the system. This chapter covers how to run commands at different times and discusses the environmental variables that affect this process.
- ❑ **Chapter 12: Security**. Unix has had security features ingrained for many years, but as with any operating system, it can be made more secure from malicious entities on the outside or inside. This chapter goes over the basics of system security and then covers some of the fundamental steps you can take to make your system more secure.

Chapters 13–17 delve into shell scripting and other methods of "automating" common tasks in Unix systems. Although these tasks often fall within the purview of system administrators, other users, including home users, may benefit.

- ❑ **Chapter 13: Basic Shell Scripting**. Shell scripting is the gateway to more advanced programming languages for many users. This chapter delves into the basics of programming with the major Unix shells, making the transition from user to beginning programmer easier.
- ❑ **Chapter 14: Advanced Shell Scripting**. This chapter takes Chapter 13 one step further, moving you into more advanced programming topics and leaving you with the capability to program shell scripts for any task.
- ❑ **Chapter 15: System Logging**. The importance of logging to users, administrators, and programmers cannot be overstated. Logging is the outlet for the system to communicate with the user, on everything from problems to successful system actions.
- ❑ **Chapter 16: Unix Networking**. This chapter covers all aspects of communicating with other systems, including network administration and scripting on common network tasks.
- ❑ **Chapter 17: Perl Programming for Unix Automation**. Perl is one of the most common programming languages on Unix, as well as on other operating systems. Perl enables you to quickly write concise, useful programs. This chapter goes over the basics of programming in the Perl language and tells you how to automate common Unix tasks with Perl.

Chapters 18 and 19 cover two important topics: backing up your data and installing Unix programs.

- ❑ **Chapter 18: Backup Tools**. This chapter describes some of the tools available on your Unix system for backing up and restoring your system in the event of accidental deletion or major system failure or catastrophe.
- ❑ **Chapter 19: Installing Software from Source Code**. Although Unix includes many programs in a default installation, there are often many other programs you will want to install. This chapter shows you how to install software from source code and from precompiled binaries.

Chapters 20 and 21 provide maps to Unix operating systems for those who are more familiar with Microsoft Windows, Microsoft DOS, Mac OS 9, and Mac OS X. These chapters are great references for those who have used other operating systems and want to compare Unix to what they already know.

- ❑ **Chapter 20: Conversion: Unix for Mac OS Users**. Mac OS X is built on a Unix foundation, but there are some minor differences between standard Unix and Apple’s Mac OS X. This chapter converts typical Mac OS (X, 9, and below) commands and concepts into their equivalent Unix commands or concepts. This chapter makes the migration into Unix much easier for users of any version of Apple’s operating systems.
- ❑ **Chapter 21: Conversion: Unix for Windows Users**. Microsoft Windows is the predominant operating system available today. This chapter converts the most common Windows and MS-DOS commands into their equivalent Unix commands, making the migration from those operating systems to Unix much simpler.

The book concludes with two appendixes. Appendix A, “Answers,” provides the solutions to the exercise(s) at the end of most chapters. These exercises will enable you to test your grasp of the concepts presented in the chapter. Appendix B, “Useful Unix Web Sites,” provides links to some of the best Unix-related Web sites on the Internet.

What Do You Need to Use This Book?

There are no requirements to use this book, but to make the learning process easier, the KNOPPIX distribution of Linux is provided on the CD-ROM accompanying this book. This enables you to use a Unix-based operating system any time, with no commitment of hard-drive resources or system alterations. The KNOPPIX distribution runs completely from CD-ROM and can be run at any time. If you have a Mac OS X system, you are already using a Unix operating system. The CD-ROM version of KNOPPIX runs only on Intel- or AMD-based systems; it will not work on Apple's line of hardware.

Conventions

This book uses the conventions discussed in this section to make the importance of specific information stand out.

> **Important notes or concepts appear in this format.**

Interesting tidbits or tips are formatted in italics, like this.

```
Code or commands are in this monotype format.
```

The text also uses specific styles to denote their significance:

- ❑ Keyboard commands that use function keys are denoted like: Shift+Q
- ❑ Web URLs are noted like this: `persistence.properties`.

Any actual Tab characters in code are represented by a right arrow: →.

Source Code

The source code for all the code in this book is available online if you prefer to cut and paste rather than copy by hand from the book. It is available at `www.wrox.com`. At the Wrox Web site, you can find the book's source code by searching for the book title (*Beginning Unix*) or ISBN (0-7645-7994-0).

Errata

This book has been checked for technical and grammatical errors, but as is human nature, errors can occur. The errata page for this book is available at `www.wrox.com`, in the book details section. If you find an error in the book that is not listed, the authors would greatly appreciate it if you go to `www.wrox.com/contact/techsupport.shtml` and complete the form to submit the error. By submitting any errors you discover, you help us to make this book even better.

1

Unix Fundamentals

The Unix operating system was created more than 30 years ago by a group of researchers at AT&T's Bell Laboratories. During the three decades of constant development that have followed, Unix has found a home in many places, from the ubiquitous mainframe to home computers to the smallest of embedded devices. This chapter provides a brief overview of the history of Unix, discusses some of the differences among the many Unix systems in use today, and covers the fundamental concepts of the basic Unix operating system.

Brief History

In terms of computers, Unix has a long history. Unix was developed at AT&T's Bell Laboratories after Bell Labs withdrew from a long-term collaboration with General Electric (G.E.) and MIT to create an operating system called MULTICS (Multiplexed Operating and Computing System) for G.E.'s mainframe. In 1969, Bell Labs researchers created the first version of Unix (then called UNICS, or Uniplexed Operating and Computing System), which has evolved into the common Unix systems of today.

Unix was gradually ported to different machine architectures from the original PDP-7 minicomputer and was used by universities. The source code was made available at a small fee to encourage its further adoption. As Unix gained acceptance by universities, students who used it began graduating and moving into positions where they were responsible for purchasing systems and software. When those people began purchasing systems for their companies, they considered Unix because they were familiar with it, spreading adoption further. Since the first days of Unix, the operating system has grown significantly, so that it now forms the backbone of many major corporations' computer systems.

Unix no longer is an acronym for anything, but it is derived from the UNICS acronym. Unix developers and users use a lot of acronyms to identify things in the system and for commands.

Unix Versions

In the early days Unix was made available as source code rather than in the typical binary form. This made it easier for others to modify the code to meet their needs, and it resulted in forks in the code, meaning that there are now many disparate versions (also known as flavors).

> *Source code represents the internal workings of a program, specifying line by line how a program or application operates. Access to source code makes it easier to understand what is occurring in the program and allows for easier modification of the program. Most commercial programs are distributed in binary form, meaning they are ready to be run, but the internal lines of code are not readable by people.*

There are primarily two base versions of Unix available: AT&T System V and Berkley Software Distribution (BSD). The vast majority of all Unix flavors are built on one of these two versions. The primary differences between the two are the utilities available and the implementations of the file structure. Most of the Unix flavors incorporate features from each base version; some include the System V version utilities in `/usr/bin` and the BSD version in `/usr/ucb/bin`, for example, so that you have the choice of using a utility with which you are comfortable. This arrangement is indicative of the Unix way of providing the flexibility to do things in different ways.

The various versions of Unix systems provide the user the power of choice: you can select the flavor that best matches your needs or system requirements. This ability to choose is considered by many as a strength, although some see it as a weakness in that these slightly differing versions and flavors create some incompatibilities (in the implementation, commands, communications, or methods, for example). There is no “true” version of Unix or one that is more official than others; there are just different implementations. Linux, for example, is a variant of Unix that was built from the ground up as a free Unix-like alternative to the expensive commercial Unix versions available when Linux was first created in 1991. Here are some of the more popular flavors of Unix available:

Sun Microsystem’s Solaris Unix

IBM AIX

Hewlett Packard HP-UX

Red Hat Enterprise Linux

Fedora Core

SUSE Linux

Debian GNU/Linux

Mac OS X

KNOPPIX

Yellow Dog Linux (for Apple systems)

Santa Cruz Operations SCO OpenServer

SGI IRIX

FreeBSD

OpenBSD

NetBSD

OS/390 Unix

Plan 9

Each of these flavors implements its version of Unix in a slightly different way, but even though the implementation of a command may vary on some systems, the core command and its functionality follow the principles of one of the two major variations. Most versions of Unix utilize SVR4 (System V) and add the BSD components as an option to allow for maximum interoperability. This is especially true with commands; for example, there are two versions of the `ps` command (for showing processes) available on most systems. One version of `ps` might reside in `/usr/bin/ps` (the System V version) while the other might exist in `/usr/ucb/bin` (BSD version); the commands operate similarly, but provide output or accept optional components in a different manner.