Unix操作系统（影印版）

5th Edition

Includes Quick Ref Card

Learning the UNIX Operating System

Jerry Peek, Grace Todino & John Strang 著

O'REILLY®

清华大学出版社

第五版

Unix操作系统（影印版）

Learning the Unix Operating System

Jerry Peek, Grace Todino & John Strang

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Paris • Sebastopol • Taipei • Tokyo
O'Reilly & Associates, Inc. 授权清华大学出版社出版

清华大学出版社

图书在版编目（CIP）数据

Unix 操作系统：/（美）彼克（Peek, J.），（美）托蒂诺（Todino, G.），（美）斯通（Strang, J.）著. —影印版. －北京：清华大学出版社，2003. 7
书名原文：Learning the Unix Operating System, Fifth Edition
ISBN 7-302-06535-7

Ⅰ.U… Ⅱ.①彼… ②托文… ③斯… Ⅲ.Unix 操作系统－英文 Ⅳ.TP316.81

中国版本图书馆 CIP 数据核字（2003）第 025949 号

北京市版权局著作权合同登记
图字：01-2003-1928 号

书　　名/ Unix 操作系统（影印版）
书　　号/ ISBN 7-302-06535-7/TP · 4902
责任编辑/ 常晓波
封面设计/ Edie Freedman，张健
出版发行/ 清华大学出版社（www.tup.tsinghua.edu.cn）
地　　址/ 北京清华大学学研大厦（邮政编码 100084）
经　　销/ 各地新华书店
印　　刷/ 北京艺辉印刷有限公司
开　　本/ 787 毫米 × 980 毫米　16 开本　11 印张
版　　次/ 2003 年 7 月第一版　2003 年 7 月第一次印刷
印　　数/ 0001-3000 册
定　　价/ 19.80 元（册）

O'Reilly & Associates公司介绍

O'Reilly & Associates公司是世界上在UNIX、X、Internet和其他开放系统图书领域具有领导地位的出版公司，同时是联机出版的先锋。

从最畅销的*The Whole Internet User's Guide & Catalog*（被纽约公共图书馆评为20世纪最重要的50本书之一）到GNN（最早的Internet门户和商业网站），再到WebSite（第一个桌面PC的Web服务器软件），O'Reilly & Associates一直处于Internet发展的最前沿。

许多书店的反馈表明，O'Reilly & Associates是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly & Associates公司具有深厚的计算机专业背景，这使得O'Reilly & Associates形成了一个非常不同于其他出版商的出版方针。O'Reilly & Associates所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly & Associates还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly & Associates依靠他们及时地推出图书。因为O'Reilly & Associates紧密地与计算机业界联系着，所以O'Reilly & Associates知道市场上真正需要什么图书。

出版说明

计算机网络与通信技术的成熟与广泛应用，以及Internet与Web的迅速发展，为人类的工业生产、商业活动和日常生活都带来了巨大的影响。网络与通信技术在我国的很多领域也已经广泛应用，并且取得了巨大的效益。然而，该领域的技术创新的速度之快也是有目共睹的。为了帮助国内技术人员和网络管理人员在第一时间掌握国外最新的技术，清华大学出版社引进了美国O'Reilly & Associates公司的一批、在计算机网络理论和Open Source方面代表前沿技术或者在某专项领域内享有盛名的著作，以飨读者。本套丛书采用影印版的形式，力求与国外图书“同步”出版，“原汁原味”地展现给读者各种权威技术理论和技术术语，适合于相关行业的高级技术人员、科研机构研究人员和高校教师阅读。

本批图书包括以下几种：

- 《802.11安全手册（影印版）》
- 《构建Internet防火墙（影印版）》
- 《Java技术手册（影印版）》
- 《Open Sources（影印版）》
- 《WWW信息体系结构（影印版）》
- 《LINUX RAID管理（影印版）》
- 《Peer-to-Peer（影印版）》
- 《Java实例技术手册（影印版）》
- 《Free As In Freedom（影印版）》
- 《Unix操作系统（影印版）》

Preface

The Unix Family of Operating Systems

An *operating system* (or "OS") is a set of programs that controls a computer. It controls both *hardware* (things you can touch, like keyboards, screens, and disk drives) and *software* (application programs that you run, like a word processor).

Some computers have a *single-user* OS, which means that only one person can use the computer at a time. Many older OSes (such as MS-DOS) can also do only one job at a time. But almost any computer can do a lot more if it has a *multiuser, multitasking* operating system such as Unix. These powerful OSes let many people use the computer at the same time and let each user run several jobs at once.

Unix was invented more than 30 years ago for scientific and professional users who wanted a very powerful and flexible OS. It's been significantly developed since then. Because Unix was designed for experts, it can be a bit overwhelming at first. But after you get the basics (from this book!) you'll start to appreciate some of the reasons to use Unix:

- It comes with a huge number of powerful application programs. You can get many others for free on the Internet. (The GNU utilities, available from the Free Software Foundation, are very popular.) You can thus do much more at a much lower cost.
- Not only are the applications often free, but some Unix versions are also free. Linux is a good example. Like the free applications, most free Unix versions are of excellent quality. They're maintained by

volunteer programmers who want a powerful OS and are frustrated by the slow, bug-ridden OS development at some big software companies.

- Unlike OSes such as Microsoft Windows and MacOS that are designed for certain types of hardware, Unix runs on almost any kind, from tiny embedded systems to giant supercomputers. After you read this book, you'll be ready to use many kinds of computers without learning a new OS for each one.
- In general, Unix (especially without a windowing system) is less resource-intensive than other major operating systems. For instance, Linux will run happily on an old system with a x386 microprocessor and let multiple users share the same computer. (Don't bother trying to use the latest versions of Microsoft Windows on a system that's more than a few years old!) If you need a windowing system, Unix lets you choose from modern feature-rich interfaces as well as from simple ones that need much less system power. Anyone with limited resources—educational institutions, organizations in developing countries, and so on—can use Unix to do more with less.
- Much of the Internet's development was done on Unix systems. Many Internet web sites and Internet service providers use Unix because it's so flexible and inexpensive. With powerful hardware, Unix really shines.

Versions of Unix

There are several versions of Unix. Until a few years ago, there were two main versions: the line of Unix releases that started at AT&T (the latest is System V Release 4), and another from the University of California at Berkeley (the last version was 4.4BSD). Some past and present commercial versions include SunOS, Solaris, SCO Unix, AIX, HP/UX, and ULTRIX. Freely available versions include Linux, NetBSD, and FreeBSD (FreeBSD is based on 4.4BSD-Lite).

Many Unix versions, including System V Release 4, merge earlier AT&T releases with BSD features. The POSIX standard for Unix-like operating systems defines a single interface to Unix. Although advanced features differ among systems, you should be able to use this introductory handbook on any system.

When we write "Unix" in this book, we mean "Unix and its versions" unless we specifically mention a particular version.

Interfaces to Unix

Unix can be used as it originally was, on typewriter-like terminals, from a shell prompt on a command line. (See the section "Examples," later in this chapter.) Most versions of Unix also work with window systems (sometimes called Graphical User Interfaces, or GUIs). These allow each user to have a single screen with multiple windows—including "terminal" windows that act like the original Unix interface. (Chapter 2 explains window system basics.)

Although a window system lets you use Unix without typing text at a shell prompt, we'll spend most of our time on that traditional command-line interface to Unix. Why?

- Every Unix system has a command-line interface. If you know how to use the command line, you'll always be able to use the system.
- If you become a more-advanced Unix user, you'll find that the command line is actually much more flexible than a windowing interface. Unix programs are designed to use together from the command line—as "building blocks"—in an almost infinite number of combinations, to do an infinite number of tasks. No windowing system that we've seen (yet!) has this tremendous power.
- You can launch and close windowing programs from the command line, but windowing programs generally can't affect a command line or programs you run from one.
- Once you learn to use the command line, you can use those same techniques to write *scripts*. These little (or big!) programs automate jobs you'd have to do manually and repetitively with a window system (unless you understand how to program a window system, which is usually a much harder job). See the section "Programming" in Chapter 8 for a brief introduction to scripting.
- In general, text-based interfaces are much easier than GUIs for sight- and hearing-impaired users.

We aren't saying that the command-line interface is right for every situation. For instance, using the Web—with its graphics and links—is usually easier with a GUI web browser. But the command line is the fundamental way to use Unix. Understanding it will let you work on any Unix system, with or without windows.

What This Handbook Covers

This book teaches basic system utility commands to get you started with Unix. Instead of overwhelming you with lots of details, we want you to be comfortable in the Unix environment as soon as possible. So we cover a command's most useful features instead of describing all its options in detail.

We also assume that your computer works properly; someone has started it, knows the procedure for turning the power off, and knows how to perform system maintenance. In other words, we don't cover Unix system administration.

Unix users can choose between many different user interfaces—shells and window systems. Our examples show the **bash** shell and the GNOME and KDE window environments. We've chosen them because they're popular and make good examples, not because we think they're always "the best." If you do advanced work or set up Unix systems for other users, we recommend learning about a variety of shells and window systems and choosing the best ones for your needs. The principles explained in this book should help you use any Unix configuration.

What's New in the Fifth Edition

Unix keeps evolving, and this book changes with it. Although most tips in this book work on all Unix systems, old and new, there have been changes since 1997 that justify a fifth edition. Over the years, readers have asked us to include topics that couldn't be covered in just a few paragraphs—a text editor, for instance. We've decided to let this little book grow just a bit by adding several-page overviews of popular Unix tools: the Pico text editor, the Pine email program, the Lynx web browser, and two interactive chat programs. Networking is much more common, so we've added a new chapter about it. Our windowing examples show newer window systems and you'll find sections about command-line editing. There's a new Glossary with definitions of common terms, and the Index has also been expanded. Finally, we've made changes suggested by our readers.

Format

The following sections describe conventions used in this handbook.

Commands

We introduce each main concept first, and then break it into task-oriented sections. Each section shows the best command to use for a task, explains what it does, and shows the syntax (how to put the command line together). The syntax is given like this:

rm *filename*

Commands appear in **boldface** type (in this example, **rm**). You should type the command exactly as it appears in the example. The variable parts (here, *filename*) will appear in *italic* type; you must supply your own value. To enter this command, you would type **rm** followed by a space and the name of the file that you want to remove, then press the RETURN key. (Your keyboard may have a key labeled ENTER or an arrow with a right-angle shaft instead of a RETURN key.) Throughout this book, the term *enter* means to type a command and press RETURN to run it.

Examples

Examples show what should happen as you enter a command. Some examples assume that you've created certain files. If you haven't, you may not get the results shown.

We use typewriter-style characters for examples. Items you type to try the example are **`boldface`**. System messages and responses are `normal text`.

Here's an example:

```
$ date
Tue Oct  9 13:39:24 MST 2001
$
```

The character "$" is the shell (system) prompt. To do this example, you would type **date** and then press RETURN. The **date** command responds "Tue Oct 9 13:39:24 MST 2001" and then returns you to the prompt.

Text you see in examples may not be exactly what you see on your screen. Different Unix versions have commands with different outputs. Sometimes we edit screen samples to eliminate distracting text or make them fit the page.

Problem Checklist

We've included a problem checklist in some sections. You may skip these parts and go back to them if you have a problem.

Exercises

Some sections have exercises to reinforce text you've read. Follow the exercises, but don't be afraid to experiment on your own.

Exercises have two columns. The lefthand column tells you what to do and the righthand column tells you how to do it. For example, a line in the section "Exercise: entering a few commands," near the end of Chapter 1, shows the following:

Get today's date	Enter **date**

To follow the exercise, type in the word **date** on your keyboard and then press the RETURN key. The lefthand column tells you what will happen.

After you try the commands, you'll have a better idea of the ones you want to learn more about. You can then get more information from a source in the section "Documentation," in Chapter 8.

Comments and Questions

Please address comments and questions concerning this book to the publisher:

O'Reilly & Associates, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
(800) 998-9938 (in the United States or Canada)
(707) 829-0515 (international or local)
(707) 829-0104 (fax)

To ask technical questions or comment on the book, send email to:

bookquestions@oreilly.com

We have a web site for the book where examples, errata, and any plans for future editions are listed. You can access this site at:

http://www.oreilly.com/catalog/lunix5/

For more information about books, conferences, Resource Centers, and the O'Reilly Network, see the O'Reilly web site at:

http://www.oreilly.com

If you write to us, please include information about your Unix environment and the computer you use. You'll have our thanks, along with thanks from future readers of this handbook.

Acknowledgments

H. Milton Peek reviewed the first draft of this edition. Jeff Kawski acted as the technical editor. Chris Stone of O'Reilly & Associates, Inc. gave information about Mac OS X and reviewed the section about it. And Tim, thanks from Jerry for all your advice and support during my 12 years of writing for O'Reilly.

Table of Contents

Preface .... *ix*

1. *Getting Started* *1*
   - Working in the Unix Environment *1*
   - Syntax of Unix Command Lines *11*
   - Types of Commands *14*
   - The Unresponsive Terminal *14*

2. *Using Window Systems* *17*
   - Introduction to Windowing *18*
   - Starting X *19*
   - Running Programs *21*
   - Working with a Mouse *26*
   - Working with Windows *29*
   - Other Window Manager Features *35*
   - Unresponsive Windows *37*
   - Other X Window Programs *39*
   - Quitting *40*

3. *Using Your Unix Account* *42*
   - The Unix Filesystem *42*
   - Looking Inside Files with less *54*
   - Protecting and Sharing Files *56*
   - Graphical Filesystem Browsers *61*

Completing File and Directory Names 62
Changing Your Password 62
Customizing Your Account 63

4. *File Management* 66
File and Directory Names 66
File and Directory Wildcards 67
Creating and Editing Files 69
Managing Your Files 74
Printing Files 81

5. *Redirecting I/O* 87
Standard Input and Standard Output 87
Pipes and Filters 92

6. *Using the Internet and Other Networks* 97
Remote Logins 97
Windows from Other Computers 100
Lynx, a Text-based Web Browser 101
Transferring Files 104
Electronic Mail 108
Usenet News 118
Interactive Chat 123

7. *Multitasking* 130
Running a Command in the Background 131
Checking on a Process 132
Cancelling a Process 134

8. *Where to Go from Here* 136
Documentation 136
Shell Aliases and Functions 139
Programming 139
Using Unix on Non-Unix Systems 140

Glossary 143

Index 149

In this chapter:

- ***Working in the Unix Environment***
- ***Syntax of Unix Command Lines***
- ***Types of Commands***
- ***The Unresponsive Terminal***

1

Getting Started

Before you can use Unix, a system staff person has to set up a Unix *account* for you. The account is identified by your *username*, which is usually a single word or an abbreviation. Think of this account as your office—it's your place in the Unix environment. Other users may also be at work on the same system. At many sites, there will be a whole network of Unix computers. So in addition to knowing your username, you may also need to know the *hostname* (name) of the computer that has your account. Alternatively, your account may be shared between all computers on the local network, and you may be able to log into any of them.

Once you've logged in to your account, you'll interact with your system by typing commands at a command line, to a program called a *shell*. You'll get acquainted with the shell, enter a few commands, and see how to handle common problems. To finish your Unix session, you'll log out.

Working in the Unix Environment

Each user communicates with the computer from a terminal. To get into the Unix environment, you first connect to the Unix computer. (Your terminal is probably already connected to a computer.* But Unix systems also let you log into other computers across a network. In this case, log into your local computer first, then use a remote login command to connect to the remote computer. See the section "Remote Logins" in Chapter 6.)

* Some terminals can connect to many computers through a kind of switchboard called a *port contender* or *data switch*. On these terminals, start by telling the port contender which computer you want to connect to.

After connecting your terminal, if needed, you start a session by logging in to your Unix account. To log in, you need your username and a *password.* Logging in does two things: it identifies which user is in a session, and it tells the computer that you're ready to start work. When you've finished, log out—and, if necessary, disconnect from the Unix computer.

If someone else has your username and password, they probably can log into your account and do anything you can. They can read private information, corrupt or delete important files, send email messages as if they came from you, and more. If your computer is connected to a network—the Internet or a local network inside your organization—intruders may also be able to log in without sitting at your keyboard! See the section "Remote Logins" in Chapter 6 for one explanation of one way this can be done.

Anyone may be able to get your username—it's usually part of your email address, for instance. Your password is what keeps others from logging in as you. Don't leave your password anywhere around your computer. Don't give your password to anyone who asks you for it unless you're sure they'll preserve your account security. Also don't send your password by email; it can be stored, unprotected, on other systems and on backup tapes, where other people may find it and then break into your account.

If you suspect that someone is using your account, ask system staff for advice. If you can't do that, setting a new password may help; see the section "Changing Your Password" in Chapter 3.

Unix systems are case sensitive. Most usernames, commands, and filenames use lowercase letters (though good passwords use a mixture of lower- and uppercase letters). Before you log in, be sure your CAPS LOCK key is off.

Connecting to the Unix Computer

If you see a message from the computer that looks something like this:

```
login:
```

you're probably connected! You can skip ahead to the section "Logging in Nongraphically" and log in.

Otherwise, if someone nearby uses the same kind of computer system you do, the easiest way to find out if you're connected is probably to ask for help. (We can't cover every user's situation exactly. There are just too many possibilities.)

If there's no one to ask, look ahead at the section "Logging in Nongraphically," later in this chapter, as well as the section "Starting X" in Chapter 2 and the section "Remote Logins" in Chapter 6. You may recognize your situation.

If that doesn't help, but your computer seems to be running an operating system other than Unix (such as Microsoft Windows), check your menus and icons for one with the name of the Unix computer you're supposed to connect to. You might also find a program named either **telnet**, **eXceed**, **ssh**, **VMware**, **procomm**, **qmodem**, **kermit**, or **minicom**, or something relating to remote access.

Logging in Nongraphically

The process of making yourself known to the computer system and getting to your Unix account is called *logging in*. If you've connected to the Unix host from another operating system, you may have been logged into Unix automatically; in this case, you should be able to run Unix programs, as shown later in this chapter in the section "Shells in a Window System" and the section "The Shell Prompt." Otherwise, before you can start work, you must connect your terminal or terminal window to the computer you need (as in the previous section) and identify yourself to the Unix system.

There are generally two ways to log in: graphically and nongraphically. If your screen has a window or windows floating in it, something like Figure 2-2A, you probably need to log in graphically, as explained by "the section "A. Ready to Run X (with a Graphical Login)" in Chapter 2.

Otherwise, to log in nongraphically, enter your username (usually your name or initials) and your private password. The password does not appear as you enter it.

When you have logged in successfully, you'll get some system messages and finally the shell prompt (where you can enter Unix commands). A successful login to the system named *nutshell* could look like Example 1-1.