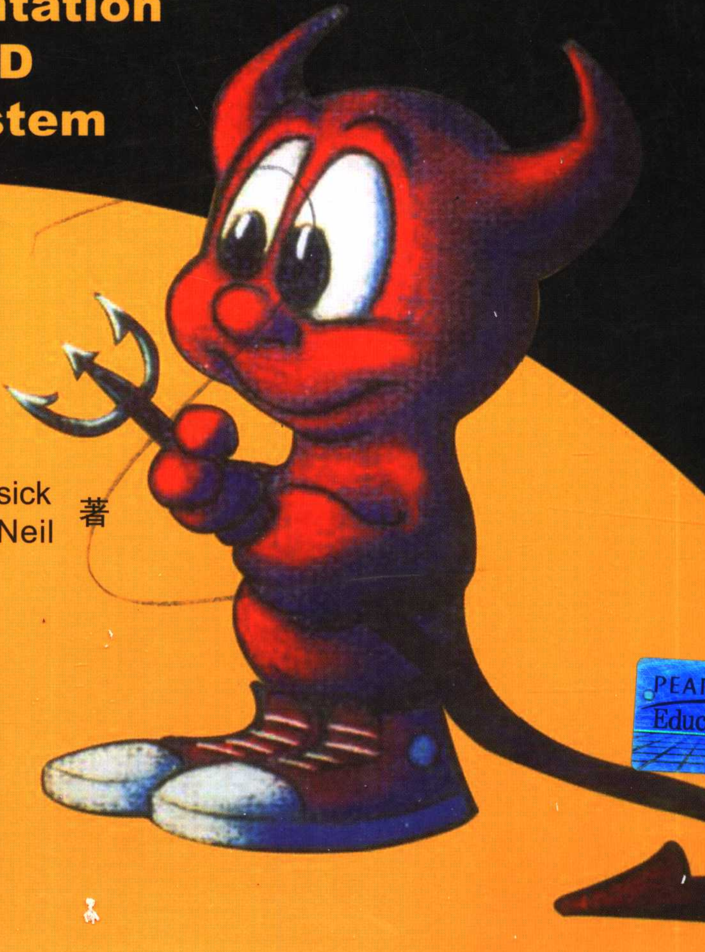
PEARSON
Addison Wesley

FreeBSD 操作系统设计与实现

（影印版）

The Design and Implementation of the FreeBSD Operating System



[美] Marshall Kirk McKusick
George V. Neville-Neil 著

中国电力出版社
CHINA ELECTRIC POWER PRESS

PEARSON Education

中国电力出版社
www.infopower.com.cn

FreeBSD

操作系统设计与实现

（[illegible]）

The Design
and Implementation
of the FreeBSD
Operating System

[illegible] 著

[illegible]

The Design and Implementation of the

FreeBSD Operating System

Marshall Kirk McKusick

George V. Neville-Neil

Addison-Wesley

Boston • San Francisco • New York • Toronto • Montreal
London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City

图书在版编目（CIP）数据

FreeBSD 操作系统设计与实现＝The Design and Implementation of FreeBSD Operating System：英文 /（美）麦卡思克，（美）尼尔著.—影印本.—北京：中国电力出版社，2008
ISBN 978-7-5083-6570-1
I. F… II.①麦…②尼… III.UNIX 操作系统－英文 IV.TP316.81
中国版本图书馆 CIP 数据核字（2007）第 200730 号

书　　名：FreeBSD 操作系统设计与实现（影印版）
编　　著：（美）Marshall Kirk McKusick, George V. Neville-Neil
责任编辑：牛贵华
出版发行：中国电力出版社
地址：北京市三里河路 6 号　　邮政编码：100044
电话：（010）58383411　　传　　真：（010）58383267
印　　刷：北京市同江印刷厂
开本尺寸：178×233　　印　　张：44.5
书　　号：ISBN 978-7-5083-6570-1
版　　次：2008 年 1 月北京第 1 版　　2008 年 1 月第 1 次印刷
印　　数：0001—3000 册
定　　价：68.00 元

Dedication

This book is dedicated to the BSD community.
Without the contributions of that community's members,
there would be nothing about which to write.

Preface

This book follows the earlier authoritative and full-length descriptions of the design and implementation of the 4.3BSD and 4.4BSD versions of the UNIX system developed at the University of California at Berkeley. Since the final Berkeley release in 1994, several groups have continued development of BSD. This book details FreeBSD, the system with the largest set of developers and the most widely distributed releases. Although the FreeBSD distribution includes nearly 1000 utility programs in its base system and nearly 10,000 optional utilities in its ports collection, this book concentrates almost exclusively on the kernel.

UNIX-like Systems

UNIX-like systems include the traditional vendor systems such as Solaris and HP-UX; the Linux-based distributions such as Red Hat, Debian, Suse, and Slackware; and the BSD-based distributions such as FreeBSD, NetBSD, OpenBSD, and Darwin. They run on computers ranging from laptops to the largest supercomputers. They are the operating system of choice for most multiprocessor, graphics, and vector-processing systems, and are widely used for the original purpose of timesharing. The most common platform for providing network services (from FTP to WWW) on the Internet, they are collectively the most portable operating system ever developed. This portability is due partly to their implementation language, C [Kernighan & Ritchie, 1989] (which is itself a widely ported language), and partly to the elegant design of the system.

Since its inception in 1969 [Ritchie & Thompson, 1978], the UNIX system has developed in several divergent and rejoining streams. The original developers continued to advance the state of the art with their Ninth and Tenth Edition UNIX inside AT&T Bell Laboratories, and then their Plan 9 successor to UNIX. Meanwhile, AT&T licensed UNIX System V as a product before selling it to Novell. Novell passed the UNIX trademark to X/OPEN and sold the source code and distribution rights to Santa Cruz Operation (SCO). Both System V and Ninth Edition

UNIX were strongly influenced by the Berkeley Software Distributions produced by the Computer Systems Research Group (CSRG) of the University of California at Berkeley. The Linux operating system, although developed independently of the other UNIX variants, implements the UNIX interface. Thus, applications developed to run on other UNIX-based platforms can be easily ported to run on Linux.

Berkeley Software Distributions

The distributions from Berkeley were the first UNIX-based systems to introduce many important features including the following:

- Demand-paged virtual-memory support
- Automatic configuration of the hardware and I/O system
- A fast and recoverable filesystem
- The socket-based interprocess-communication (IPC) primitives
- The reference implementation of TCP/IP

The Berkeley releases found their way into the UNIX systems of many vendors and were used internally by the development groups of many other vendors. The implementation of the TCP/IP networking protocol suite in 4.2BSD and 4.3BSD, and the availability of those systems, played a key role in making the TCP/IP networking protocol suite a world standard. Even the non-UNIX vendors such as Microsoft have adopted the Berkeley socket design in their Winsock IPC interface.

The BSD releases have also been a strong influence on the POSIX (IEEE Std 1003.1) operating-system interface standard, and on related standards. Several features—such as reliable signals, job control, multiple access groups per process, and the routines for directory operations—have been adapted from BSD for POSIX.

Early BSD releases contained licensed UNIX code, thus requiring recipients to have an AT&T source license to be able to obtain and use BSD. In 1988, Berkeley separated its distribution into AT&T licensed and freely redistributable code. The freely redistributable code was licensed separately and could be obtained, used, and redistributed by anyone. The final freely redistributable 4.4BSD-Lite2 release from Berkeley in 1994 contained nearly the entire kernel and all the important libraries and utilities.

Two groups, NetBSD and FreeBSD, sprang up in 1993 to begin supporting and distributing systems built from the freely redistributable releases being done by Berkeley. The NetBSD group emphasized portability and the minimalist approach, porting the systems to nearly forty platforms and pushing to keep the system lean to aid embedded applications. The FreeBSD group emphasized maximal support for the PC architecture and pushed to ease installation for, and market their system to, as wide an audience as possible. In 1995, the OpenBSD group split from the NetBSD group to develop a distribution that emphasized security. Over the years there has been a healthy competition among the BSD distributions, with many ideas and much code flowing between them.

Material Covered in this Book

This book is about the *internal* structure of the FreeBSD 5.2 kernel and about the concepts, data structures, and algorithms used in implementing FreeBSD's system facilities. Its level of detail is similar to that of Bach's book about UNIX System V [Bach, 1986]; however, this text focuses on the facilities, data structures, and algorithms used in the FreeBSD variant of the UNIX operating system. The book covers FreeBSD from the system-call level down—from the interface to the kernel to the hardware itself. The kernel includes system facilities, such as process management, virtual memory, the I/O system, filesystems, the *socket* IPC mechanism, and network protocol implementations. Material above the system-call level—such as libraries, shells, commands, programming languages, and other user interfaces—is excluded, except for some material related to the terminal interface and to system startup. Following the organization first established by Organick's book about Multics [Organick, 1975], this book is an in-depth study of a contemporary operating system.

Where particular hardware is relevant, the book refers to the Intel Personal Computer (PC) architecture. Because FreeBSD has emphasized development on the PC, that is the architecture with the most complete support, so it provides a convenient point of reference.

Use by Computer Professionals

FreeBSD is widely used to support the core infrastructure of many companies worldwide. Because it can be built with a small footprint, it is also seeing increased use in embedded applications. The licensing terms of FreeBSD do not require the distribution of changes and enhancements to the system. The licensing terms of Linux require that all changes and enhancements to the kernel be made available in source form at minimal cost. Thus, companies that need to control the distribution of their intellectual property build their products using FreeBSD.

This book is of direct use to the professionals who work with FreeBSD systems. Individuals involved in technical and sales support can learn the capabilities and limitations of the system; applications developers can learn how to effectively and efficiently interface to the system; system administrators without direct experience with the FreeBSD kernel can learn how to maintain, tune, and configure the system; and systems programmers can learn how to extend, enhance, and interface to the system.

Readers who will benefit from this book include operating-system implementors, system programmers, UNIX application developers, administrators, and curious users. The book can be read as a companion to the source code of the system, falling as it does between the manual pages and the code in detail of treatment. But this book is neither exclusively a UNIX programming manual nor a user tutorial (for a tutorial, see Libes & Ressler [1988]). Familiarity with the use of some version of the UNIX system (see, for example, Stevens [1992]) and with the C programming language (see, for example, Kernighan & Ritchie [1989]) would be extremely useful.

Use in Courses on Operating Systems

This book is suitable for use as a reference text to provide background for a primary textbook in a first-level course on operating systems. It is not intended for use as an introductory operating-system textbook; the reader should have already encountered terminology such as *memory management*, *process scheduling*, and *I/O systems* [Silberschatz et al., 2002]. Familiarity with the concepts of network protocols [Comer, 2000; Stallings, 2000; Tanenbaum, 2003] will be useful for understanding some of the later chapters.

This book can be used in combination with a copy of the FreeBSD system for more advanced operating systems courses. Students' assignments can include changes to, or replacements of, key system components such as the scheduler, the paging daemon, the filesystems, thread signalling, various networking layers, and I/O management. The ability to load, replace, and unload modules from a running kernel allows students to experiment without the need to compile and reboot the system. By working with a real operating system, students can directly measure and experience the effects of their changes. Because of the intense peer review and insistence on well-defined coding standards throughout its 25-year lifetime, the FreeBSD kernel is considerably cleaner, more modular, and thus easier to understand and modify than most software projects of its size and age.

Exercises are provided at the end of each chapter. The exercises are graded into three categories indicated by zero, one, or two asterisks. The answers to exercises that carry no asterisks can be found in the text. Exercises with a single asterisk require a step of reasoning or intuition beyond a concept presented in the text. Exercises with two asterisks present major design projects or open research questions.

Organization

This text discusses both philosophical and design issues, as well as details of the actual implementation. Often, the discussion starts at the system-call level and descends into the kernel. Tables and figures are used to clarify data structures and control flow. Pseudocode similar to the C language displays algorithms. Boldface font identifies program names and filesystem pathnames. Italics font introduces terms that appear in the glossary and identifies the names of system calls, variables, routines, and structure names. Routine names (other than system calls) are further identified by the name followed by a pair of parenthesis (e.g., *malloc*() is the name of a routine, whereas *argv* is the name of a variable).

The book is divided into five parts, organized as follows:

- **Part I, Overview** Three introductory chapters provide the context for the complete operating system and for the rest of the book. Chapter 1, *History and Goals*, sketches the historical development of the system, emphasizing the system's research orientation. Chapter 2, *Design Overview of FreeBSD*, describes the services offered by the system and outlines the internal organization of the kernel. It also discusses the design decisions that were made as the system was developed. Sections 2.3 through 2.14 in Chapter 2 give an overview of their corresponding chapter. Chapter 3, *Kernel Services*, explains how system calls are done and describes in detail several of the basic services of the kernel.

- **Part II, Processes** The first chapter in this part—Chapter 4, *Process Management*—lays the foundation for later chapters by describing the structure of a process, the algorithms used for scheduling the execution of the threads that make up a process, and the synchronization mechanisms used by the system to ensure consistent access to kernel-resident data structures. In Chapter 5, *Memory Management*, the virtual-memory-management system is discussed in detail.
- **Part III, I/O System** First, Chapter 6, *I/O System Overview*, explains the system interface to I/O and describes the structure of the facilities that support this interface. Following this introduction are four chapters that give the details of the main parts of the I/O system. Chapter 7, *Devices*, gives a description of the I/O architecture of the PC and describes how the I/O subsystem is managed and how the kernel initially maps out and later manages the arrival and departure of connected devices. Chapter 8, *Local Filesystems*, details the data structures and algorithms that implement filesystems as seen by application programs as well as how local filesystems are interfaced with the device interface described in Chapter 7. Chapter 9, *The Network Filesystem*, explains the network filesystem from both the server and client perspectives. Chapter 10, *Terminal Handling*, discusses support for character terminals and provides a description of the pseudo-terminal device driver.
- **Part IV, Interprocess Communication** Chapter 11, *Interprocess Communication*, describes the mechanism for providing communication between related or unrelated processes. Chapters 12 and 13, *Network Communication* and *Network Protocols*, are closely related because the facilities explained in the former are implemented by specific protocols, such as the TCP/IP protocol suite, explained in the latter.
- **Part V, System Operation** Chapter 14, *Startup and Shutdown*, discusses system startup and shutdown and explains system initialization at the process level from kernel initialization to user login.

The book is intended to be read in the order that the chapters are presented, but the parts other than Part I are independent of one another and can be read separately. Chapter 14 should be read after all the others, but knowledgeable readers may find it useful independently.

At the end of the book are a Glossary with brief definitions of major terms and an Index. Each chapter contains a Reference section with citations of related material.

Getting BSD

All the BSD distributions are available either for downloading from the net or on removable media such as CD-ROM or DVD. Information on obtaining source and binaries for FreeBSD can be obtained from http://www.FreeBSD.org. The NetBSD distribution is compiled and ready to run on most workstation architectures. For more information, contact the NetBSD Project at http://www.NetBSD.org/. The OpenBSD distribution is compiled and ready to run on a wide variety of workstation architectures and has been extensively vetted for security and reliability. For more information, visit the OpenBSD project's Web site at http://www.OpenBSD.org/.

For you diehards who actually read to the end of the preface, your reward is finding out that you can get T-shirts that are a reproduction of the original artwork drawn by John Lasseter for the cover of this book (yes, he is *the* John Lasseter of Pixar fame who masterminded the production of *Toy Story*). For further information on purchasing a shirt, visit the "History of BSD T-shirts" Web page at http://www.McKusick.com/beastie/. Other items available for sale on the site include a 32-hour introductory video course based on this book, a 40-hour advanced video course based on the FreeBSD 5.2 source code, a 2.5-hour video lecture on the history of BSD, and a 4-CD set containing all the releases and the source-control history of BSD from Berkeley. These items are described in the advertisements that follow the Index.

Acknowledgments

We extend special thanks to Nate Lawson, who provided invaluable insight on the workings of the PC architecture and FreeBSD's interface to it.

We also thank the following people who read and commented on nearly the entire book: Michael Schuster (Sun Microsystems, Inc.) and our Addison-Wesley reviewers Chris Cooper (Hewlett-Packard) and Robert Kitzberger (IBM).

We thank the following people, all of whom read and commented on early drafts of various chapters of the book: Samy Al Bahra (Kerneled.com), Dorr H. Clark (Santa Clara University), Matthew Dillon (The DragonFly BSD Project), John Dyson, Andreas Gustafsson, Poul-Henning Kamp (The FreeBSD Project), David G. Lawrence (The FreeBSD Project), Samuel Leffler, M. Warner Losh (Timing Solutions), Andre Oppermann (Internet Business Solutions AG), David I. Parfitt (independent hacker), Doug Rabson (Qube Software Ltd.), Jeffrey Roberson (The FreeBSD Project), Soren Schmidt (FreeBSD senior developer), Ken Smith (University at Buffalo CSE Department), Gregory Sutter (*Daemon News*), Charles P. Wright (SUNY Stony Brook), and Erez Zadok (Stony Brook University).

We are grateful to our editor, Peter Gordon, who had faith in our ability to get the book written despite several years of delays on our part and who accelerated the production when we finally had a completed manuscript. We thank all the professional people at Addison-Wesley who helped us bring the book to completion: managing editor John Fuller, editorial assistant Bernie Gaffney, production supervisor Elizabeth Ryan, and cover designer Chuti Prasertsith. We also thank the people at Stratford Publishing Services: manager of editorial services Kathy Glidden, copy editor Debbie Prato, and proofreader Hilary Farquhar. Finally we acknowledge the contributions of Jaap Akkerhuis, who designed the troff macros for the BSD books, and John Lasseter, who drew the original BSD daemon art used on the cover.

This book was produced using James Clark's implementations of **pic**, **tbl**, **eqn**, and **groff**. The index was generated by **awk** scripts derived from indexing programs written by Jon Bentley and Brian Kernighan [Bentley & Kernighan, 1986]. Most of the art was created with **xfig**. Figure placement and widow elimination were handled by the **groff** macros, but orphan elimination and production of even page bottoms had to be done by hand.

We encourage readers to send us suggested improvements or comments about typographical or other errors found in the book; please send electronic mail to **FreeBSDbook-bugs@McKusick.COM**.

References

Bach, 1986.

M. J. Bach, *The Design of the UNIX Operating System,* Prentice-Hall, Englewood Cliffs, NJ, 1986.

Bentley & Kernighan, 1986.

J. Bentley & B. Kernighan, "Tools for Printing Indexes," Computing Science Technical Report 128, AT&T Bell Laboratories, Murray Hill, NJ, 1986.

Comer, 2000.

D. Comer, *Internetworking with TCP/IP Volume 1,* 4th ed., Prentice-Hall, Upper Saddle River, NJ, 2000.

Kernighan & Ritchie, 1989.

B. W. Kernighan & D. M. Ritchie, *The C Programming Language,* 2nd ed., Prentice-Hall, Englewood Cliffs, NJ, 1989.

Libes & Ressler, 1988.

D. Libes & S. Ressler, *Life with UNIX,* Prentice-Hall, Englewood Cliffs, NJ, 1988.

Organick, 1975.

E. I. Organick, *The Multics System: An Examination of Its Structure,* MIT Press, Cambridge, MA, 1975.

Ritchie & Thompson, 1978.

D. M. Ritchie & K. Thompson, "The UNIX Time-Sharing System," *Bell System Technical Journal,* vol. 57, no. 6, Part 2, pp. 78–90, July–August 1978. The original version [*Comm. ACM vol. 7, no. 7, pp. 365–375 (July 1974)] described the 6th edition; this citation describes the 7th edition.*

Silberschatz et al., 2002.

A. Silberschatz, P. Galvin, & G. Gagne, *Operating System Concepts,* 6th ed., John Wiley and Sons, Hoboken, NJ, 2002.

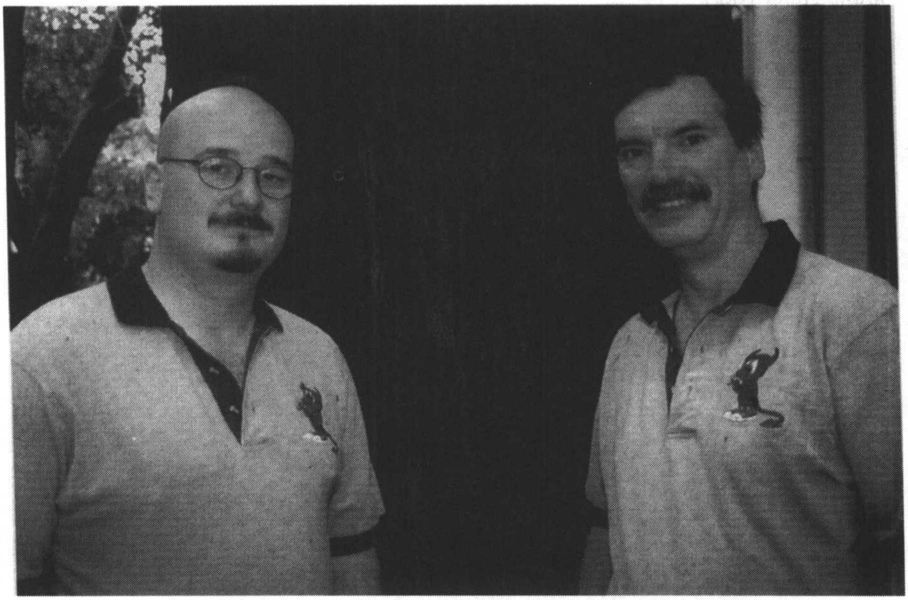
Stallings, 2000.

R. Stallings, *Data and Computer Communications,* 6th ed., Prentice Hall, Hoboken, NJ, 2000.

Stevens, 1992.
W. Stevens, *Advanced Programming in the Unix Environment,* Addison-Wesley, Reading, MA, 1992.

Tanenbaum, 2003.
A. S. Tanenbaum, *Computer Networks,* 4th ed., Prentice-Hall, Englewood Cliffs, NJ, 2003.

About the Authors



George V. Neville-Neil (left) and Marshall Kirk McKusick (right).

Marshall Kirk McKusick writes books and articles, consults, and teaches classes on UNIX- and BSD-related subjects. While at the University of California at Berkeley, he implemented the 4.2BSD fast file system and was the Research Computer Scientist at the Berkeley Computer Systems Research Group (CSRG), overseeing the development and release of 4.3BSD and 4.4BSD. His particular areas of interest are the virtual-memory system and the filesystem. He earned his undergraduate degree in electrical engineering from Cornell University and did his graduate work at the University of California at Berkeley, where he received master's degrees in computer science and business administration and a doctoral degree in computer science. He has twice been president of the board of the Usenix Association, is currently a member of the editorial board of ACM's *Queue* magazine, and is a member of the Usenix Association, ACM, and IEEE. In his spare time, he enjoys swimming, scuba diving, and wine collecting. The wine is stored in a specially constructed wine cellar (accessible from the Web at http://www.McKusick.com/cgi-bin/readhouse) in the basement of the house that he shares with Eric Allman, his domestic partner of 25-and-some-odd years.

George V. Neville-Neil works on networking and operating system code for fun and profit and also teaches courses on various subjects related to programming. His areas of interest are code spelunking, real-time operating systems, and networking. He earned his bachelor's degree in computer science at Northeastern University in Boston, Massachusetts. He serves on the editorial board of ACM's *Queue* magazine and is a member of the Usenix Association, ACM, and IEEE. He is an avid bicyclist, motorcyclist, and traveler who has made San Francisco his home since 1990.

Contents