

国外著名高等院校
信息科学与技术优秀教材

Addison
Wesley

4.4BSD

操作系统设计与实现

The Design and Implementation of the
4.4BSD Operating System

Marshall Kirk McKusick

Keith Bostic

Michael J. Karels

John S. Quarterman



John Lasseter 94

英文版

人民邮电出版社

POSTS & TELECOMMUNICATIONS PRESS

国外著名高等院校信息科学与技术优秀教材

4.4BSD 操作系统设计与实现

(英文版)

**The Design and Implementation of
the 4.4 BSD Operating System**

Marshall Kirk McKusick

Keith Bostic

Michael J. Karels

John S. Quarterman

江苏工业学院图书馆
藏书章

人民邮电出版社

国外著名高等院校信息科学与技术优秀教材

4. 4BSD 操作系统设计与实现 (英文版)

- ◆ 著 Marshall Kirk McKusick Keith Bostic
Michael J.Karels John S.Quarterman
责任编辑 陈冀康

- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
读者热线 010-67180876
北京汉魂图文设计有限公司制作
北京朝阳展望印刷厂印刷
新华书店总店北京发行所经销

- ◆ 开本: 800×1000 1/16
印张: 38
字数: 703 千字 2002 年 9 月第 1 版
印数: 1-3 000 册 2002 年 9 月北京第 1 次印刷

著作权合同登记 图字: 01 - 2001 - 3750 号

ISBN 7-115-10348-8/TP • 2907

定价: 48.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

内 容 提 要

本书是对 4.4BSD 操作系统的设计和实现的全面描述，在讨论设计理论的同时，也注重介绍实现的细节。

全书分为五个部分，共 14 章。第一部分介绍了 BSD 的历史，简单介绍了操作系统设计的几个基本方面，还介绍了内核服务的基本内容。第二部分介绍了进程管理和内存管理的内容。第三部分介绍 I/O 系统，包括本地和网络文件系统以及终端处理。第四部分介绍了进程间通信、网络通信以及网络协议。第五部分介绍了系统操作知识。

本书适合作为高等院校计算机专业操作系统课程教材，也适合开放系统下的管理人员和开发人员阅读参考。

出版说明

2001年，教育部印发了《关于“十五”期间普通高等教育教材建设与改革的意见》。该文件明确指出，“九五”期间原国家教委在“抓好重点教材，全面提高质量”方针指导下，调动了各方面的积极性，产生了一大批具有改革特色的新教材。然而随着科学技术的飞速发展，目前高校教材建设工作仍滞后于教学改革的实践，一些教材内容陈旧，不能满足按新的专业目录修订的教学计划和课程设置的需要。为此该文件明确强调，要加强国外教材的引进工作。当前，引进的重点是信息科学与技术 and 生物科学与技术两大学科的教材。要根据专业（课程）建设的需要，通过深入调查、专家论证，引进国外优秀教材。要注意引进教材的系统配套，加强对引进教材的宣传，促进引进教材的使用和推广。

邓小平同志早在1977年就明确指出：“要引进外国教材，吸收外国教材中有益的东西。”随着我国加入WTO，信息产业的国际竞争将日趋激烈，我们必须尽快培养出大批具有国际竞争能力的高水平信息技术人才。教材是一个很关键的问题，国外的一些优秀教材不但内容新，而且还提供了很多新的研究方法和思考方式。引进国外原版教材，可以促进我国教学水平的提高，提高学生的英语水平和学习能力，保证我们培养出的学生具有国际水准。

为了贯彻中央“科教兴国”的方针，配合国内高等教育教材建设的需要，人民邮电出版社约请有关专家反复论证，与国外知名的教材出版公司合作，陆续引进一些信息科学与技术优秀教材。第一批教材针对计算机专业的主干核心课程，是国外著名高等院校所采用的教材，教材的作者都是在相关领域享有盛名的专家教授。这些教材内容新，反映了计算机科学技术的最新发展，对全面提高我国信息科学与技术的教学水平必将起到巨大的推动作用。

出版国外著名高等院校信息科学与技术优秀教材的工作将是一个长期的、坚持不懈的过程，我社网站（www.ptpress.com.cn）上介绍了我们首批陆续推出的图书的详细情况，后续教材的引进和出版情况我们会及时在网上发布，敬请关注。希望广大教师和学生将使用中的意见和建议及时反馈给我们，我们将根据您的反馈不断改进我们的工作，推出更多更好的引进版信息科学与技术教材。

人民邮电出版社
2001年12月

序

从 1969 年 Bell 实验室开发出第一版 UNIX 系统至今已经有 30 多年的时间了。1976 年, Bell 实验室以外第一个可用的 UNIX 分时系统第六版(即 V6)正式发布。在此后 20 多年的时间里, UNIX 系统得到了长足的发展, 各种研究开发机构和大学都开发了不同版本的 UNIX 操作系统; 其中由加利福尼亚大学伯克利分校研究开发的 BSD (Berkeley Software Distributions) 系统是最具影响的系统之一。先后有 1BSD、2BSD、3BSD、4BSD 等, 特别是 4BSD, 又有 4.0BSD、4.1BSD、4.2BSD、4.3BSD 和 4.4BSD 等不同的版本。4.4BSD 不仅可以运行在原来的 VAX 计算机上, 而且已可以运行在 68000、SPARC、MIPS、Intel PC 等体系结构的计算机上。

The Design and Implementation of the 4.4 BSD Operating System 讨论了 4.4BSD 操作系统的设计和实现, 主要内容分为五部分共 14 章, 各章的主要内容如下:

第一部分: 总论

第 1 章 历史与目标。主要介绍 UNIX 和 4.4BSD 操作系统的历史和设计目标。

第 2 章 4.4BSD 的设计概述。描述了系统提供的服务及内核的组织概况, 还讨论了系统开发过程中的设计决策。

第 3 章 内核服务。描述了系统如何调用和工作, 详细描述了内核的几种基本服务。

第二部分: 进程

第 4 章 进程管理。描述了进程结构、进程调度算法, 以及系统为保证常驻内核的数据结构能够被正常访问所采用的同步机制, 为后面的章节奠定基础。

第 5 章 存储管理。对虚拟存储管理进行了详细讨论。

第三部分: I/O 系统

第 6 章 I/O 系统概述。说明了系统 I/O 接口以及支持这些 I/O 接口的设备结构。

第 7 章 本地文件系统。从应用程序的角度详细叙述了文件系统实现过程中用到的数据结构和算法。

第 8 章 本地文件存储。描述了本地文件系统如何与本地介质接口。

第 9 章 网络文件系统。从服务器和客户机的视角分析网络文件系统。

第 10 章 面向字符的终端的处理。讨论对字符终端的支持以及面向对象字符终端的设备驱动。

第四部分：进程通信

第 11 章 进程通信。描述了相关进程之间的通信机制。

第 12 章 网络通信。

第 13 章 网络协议。

这两章密切相关，前者所提到的设备通过后者所提到的具体协议来实现。

第五部分：系统操作

第 14 章 系统启动。讨论系统启动、关闭和配置，从进程级来讨论系统初始化，区别于对用户登录时内核初始化的讨论。

本书的四位作者都参加过 UNIX、BSD 系统的开发、教学、咨询等各种活动，亲身经历了该系统的发展过程，熟悉系统内核，具有丰富的实践经验。因此，他们在书中对 UNIX 发展历程、各个阶段的版本的设计目标和特点做出了简明和清晰的概括；对 4.4BSD 的核心设计思想和技术特征进行了准确的描述，特别是对 4.4BSD 的系统调用、中断处理、进程管理、处理机调度、虚拟存储管理、I/O 系统和设备处理、文件系统、进程间通信、网络通信、网络协议（尤其是 TCP/IP 协议的实现）等都进行了不同程度的讨论。对于对操作系统的基本原理已有一个全面了解的读者而言，本书是对基本原理的一个实现实例的全面分析，使读者可以生动地理解到操作系统的基本原理是如何体现在一个真实的操作系统中的。

1989 年，本书的前身 *The Design and Implementation of the 4.3BSD UNIX Operating System* 问世，成为其时第一本介绍 BSD 操作系统内核的技术图书，一直是各技术网站、操作系统专家和海外学子们热衷推荐阅读的图书。遗憾的是，那时候国内的读者接触英文原版图书的机会很少。随着 BSD 系统的发展，1996 年专门针对的 4.4BSD 的这本书出版了。在本书中，增加了新版本的技术信息，从系统结构出发的独特视角、符合逻辑的认识方法、清晰详尽的分析过程等特色都得到了很好的延继；因此，本书在专业技术领域的全面、权威、及时的形象和地位得到了巩固。现在，人民邮电出版社以影印方式把本书引入国内，将会给计算机专业的高年级本科生和研究生提供一个很有实用价值的 UNIX 操作系统教材。对于分布操作系统领域的研究和开发人员，本书也有很好的参考价值。

清华大学计算机系教授 博导



Preface

This book is an extensive revision of the first authoritative and full-length description of the design and implementation of the research versions of the UNIX system developed at the University of California at Berkeley. Most detail is given about 4.4BSD, which incorporates the improvements of the previous Berkeley versions. Although 4.4BSD includes nearly 500 utility programs in addition to the kernel, this book concentrates almost exclusively on the kernel.

The UNIX System

The UNIX system runs on computers ranging from personal home systems to the largest supercomputers. It is the operating system of choice for most multiprocessor, graphics, and vector-processing systems, and is widely used for its original purpose of timesharing. It is the most common platform for providing network services (from FTP to WWW) on the Internet. It is the most portable operating system ever developed. This portability is due partly to its implementation language, C [Kernighan & Ritchie, 1978] (which is itself one of the most widely ported languages), and partly to the elegant design of the system. Many of the system's features are imitated in other systems [O'Dell, 1987].

Since its inception in 1969 [Ritchie & Thompson, 1978], the UNIX system has developed in a number of 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.

Berkeley Software Distributions

These Berkeley systems have introduced several useful programs and facilities to the UNIX community:

- 2BSD (the Berkeley PDP-11 system): the text editor `vi`
- 3BSD (the first Berkeley VAX system): demand-paged virtual-memory support
- 4.0BSD: performance improvements
- 4.1BSD: job control, autoconfiguration, and long C identifiers
- 4.2BSD and 4.3BSD: reliable signals; a fast filesystem; improved networking, including a reference implementation of TCP/IP; sophisticated interprocess-communication (IPC) primitives; and more performance improvements
- 4.4BSD: a new virtual memory system; a stackable and extensible vnode interface; a network filesystem (NFS); a log-structured filesystem, numerous filesystem types, including loopback, union, and uid/gid mapping layers; an ISO9660 filesystem (e.g., CD-ROM); ISO networking protocols; support for 68K, SPARC, MIPS, and PC architectures; POSIX support, including termios, sessions, and most utilities; multiple IP addresses per interface; disk labels; and improved booting

4.2BSD, 4.3BSD, and 4.4BSD are the bases for the UNIX systems of many vendors, and are used internally by the development groups of many other vendors. Many of these developments have also been incorporated by System V, or have been added by vendors whose products are otherwise based on System V.

The implementation of the TCP/IP networking protocol suite in 4.2BSD and 4.3BSD, and the availability of those systems, explain why the TCP/IP networking protocol suite is implemented so widely throughout the world. Numerous vendors have adapted the Berkeley networking implementations, whether their base system is 4.2BSD, 4.3BSD, 4.4BSD, System V, or even Digital Equipment Corporation's VMS or Microsoft's Winsock interface in Windows '95 and Windows/NT.

4BSD has 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 4.3BSD for POSIX.

Material Covered in this Book

This book is about the *internal* structure of 4.4BSD [Quarterman et al, 1985], and about the concepts, data structures, and algorithms used in implementing 4.4BSD'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 Berkeley variant of the UNIX operating system. The book covers 4.4BSD 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. Like 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 Hewlett-Packard HP300 (Motorola 68000-based) architecture. Because 4.4BSD was developed on the HP300, that is the architecture with the most complete support, so it provides a convenient point of reference.

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 [CSRG, 1994] and the code in detail of treatment. But this book is specifically neither 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, [Kernighan & Pike, 1984]), and with the C programming language (see, for example, [Kernighan & Ritchie, 1988]) 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 second-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 & Galvin, 1994]. Familiarity with the concepts of network protocols [Tanenbaum, 1988; Stallings, 1993; Schwartz, 1987] will be useful for understanding some of the later chapters.

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 is used to display 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 1, 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 4.4BSD*, 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 2, 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 processes, 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 3, 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, *Local Filesystems*, details the data structures and algorithms that implement filesystems as seen by application programs. Chapter 8, *Local Filestores*, describes how local filesystems are interfaced with local media. 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 a character-oriented device driver.
- **Part 4, 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, as the facilities explained in the former are implemented by specific protocols, such as the TCP/IP protocol suite, explained in the latter.
- **Part 5, System Operation** Chapter 14, *System Startup*, discusses system startup, shutdown, and configuration, 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 1 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 4.4BSD

Current information about the availability of 4.4BSD source code can be found at the sites listed below. At press time, the source code for the 4.4BSD-Lite Release 2 system, as well as that for the FreeBSD version of 4.4BSD, which is compiled and ready to run on PC-compatible hardware, are available from Walnut Creek CDROM. Contact Walnut Creek for more information at 1-800-786-9907, or use orders@cdrom.com, or <http://www.cdrom.com/>. The NetBSD distribution is compiled and ready to run on most workstation architectures. For more information, contact the NetBSD Project at majordomo@NetBSD.ORG (send a message body of "lists"), or <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/>. A fully supported commercial release, BSD/OS, is available from Berkeley Software Design, Inc., at 1-800-800-4273, bsdi-info@bsdi.com, or <http://www.bsdi.com/>. The 4.4BSD manuals are jointly published by Usenix and O'Reilly. O'Reilly sells the five volumes individually or in a set (ISBN 1-56592-082-1): 1-800-889-8969, order@ora.com, or <http://www.ora.com/>.

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 the original artwork drawn by John Lasseter for the cover of this book (yes, he is the John Lasseter of Walt Disney/Pixar fame who masterminded the production of "Toy Story"). These shirts were made available to the people who helped with the creation, reviewing, and editing of the book and to those folks who first reported errors in the book. A variation on these shirts that is clearly different from the originals (so as not to diminish the rarity of the ones that people had to work to get) is now available. For further information on purchasing a shirt, send a self-addressed envelope (United States residents please include return postage) to

M. K. McKusick
1614 Oxford St.
Berkeley, CA 94709-1608

Alternatively, you can visit the "History of BSD T-shirts" web page at <http://www.mckusick.com/beastie/index.html>.

Acknowledgments

We extend special thanks to Mike Hibler (University of Utah) who coauthored Chapter 5 on memory management, and to Rick Macklem (University of Guelph), whose NFS papers provided much of the material on NFS for Chapter 9.

We also thank the following people who read and commented on nearly the entire book: Paul Abrahams (Consultant), Susan LoVerso (Orca Systems), George Neville-Neil (Wind River Systems), and Steve Stepanek (California State University, Northridge).

We thank the following people, all of whom read and commented on early drafts of the book: Eric Allman (Pangaea Reference Systems), Eric Anderson

(University of California at Berkeley), Mark Andrews (Alias Research), Mike Beede (Secure Computing Corporation), Paul Borman (Berkeley Software Design), Peter Collinson (Hillside Systems), Ben Cottrell (NetBSD user), Patrick Cua (De La Salle University, Philippines), John Dyson (The FreeBSD Project), Sean Eric Fagan (BSD developer), Mike Fester (Medieus Systems Corporation), David Greenman (The FreeBSD Project), Wayne Hathaway (Auspex Systems), John Heidemann (University of California at Los Angeles), Jeff Honig (Berkeley Software Design), Gordon Irlam (Cygnus Support), Alan Langerman (Orca Systems), Sam Leffler (Silicon Graphics), Casimir Lesiak (NASA/Ames Research Center), Gavin Lim (De La Salle University, Philippines), Steve Lucco (Carnegie Mellon University), Jan-Simon Pendry (Sequent, UK), Arnold Robbins (Georgia Institute of Technology), Peter Salus (UNIX historian), Wayne Sawdon (Carnegie Mellon University), Margo Seltzer (Harvard University), Keith Sklower (University of California at Berkeley), Keith Smith (Harvard University), and Humphrey C. Sy (De La Salle University, Philippines).

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 bsdbook-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.

CSRG, 1994.

CSRG, in *4.4 Berkeley Software Distribution*, O'Reilly & Associates, Inc., Sebastopol, CA, 1994.

Kernighan & Pike, 1984.

B. W. Kernighan & R. Pike, *The UNIX Programming Environment*, Prentice-Hall, Englewood Cliffs, NJ, 1984.

Kernighan & Ritchie, 1978.

B. W. Kernighan & D. M. Ritchie, *The C Programming Language*, Prentice-Hall, Englewood Cliffs, NJ, 1978.

Kernighan & Ritchie, 1988.

B. W. Kernighan & D. M. Ritchie, *The C Programming Language*, 2nd ed, Prentice-Hall, Englewood Cliffs, NJ, 1988.

Libes & Ressler, 1988.

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

O'Dell, 1987.

M. O'Dell, "UNIX: The World View," *Proceedings of the 1987 Winter USENIX Conference*, pp. 35–45, January 1987.

Organick, 1975.

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

Quarterman et al, 1985.

J. S. Quarterman, A. Silberschatz, & J. L. Peterson, "4.2BSD and 4.3BSD as Examples of the UNIX System," *ACM Computing Surveys*, vol. 17, no. 4, pp. 379–418, December 1985.

Ritchie & Thompson, 1978.

D. M. Ritchie & K. Thompson, "The UNIX Time-Sharing System," *Bell System Technical Journal*, vol. 57, no. 6, Part 2, pp. 1905–1929, 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.

Schwartz, 1987.

M. Schwartz, *Telecommunication Networks*, Series in Electrical and Computer Engineering, Addison-Wesley, Reading, MA, 1987.

Silberschatz & Galvin, 1994.

A. Silberschatz & P. Galvin, *Operating System Concepts*, 4th Edition, Addison-Wesley, Reading, MA, 1994.

Stallings, 1993.

R. Stallings, *Data and Computer Communications*, 4th Edition, Macmillan, New York, NY, 1993.

Tanenbaum, 1988.

A. S. Tanenbaum, *Computer Networks*, 2nd ed, Prentice-Hall, Englewood Cliffs, NJ, 1988.

About the Authors



Left to right: Mike Karels, Keith Bostic, Kirk McKusick, and John Quarterman together for the first time at a Usenix Conference in San Diego.

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. One day, he hopes to see them merged seamlessly. 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 Masters degrees in Computer Science and Business Administration, and a doctoral degree in Computer Science. He is a past president of the Usenix Association, and is a member of 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://winecellar.mckusick.com/cgi-bin/readhouse> or using “telnet winecellar.McKusick.COM 451”) in the basement of the house that he shares with Eric Allman, his domestic partner of 17-and-some-odd years.

Keith Bostic is a member of the technical staff at Berkeley Software Design, Inc. He spent 8 years as a member of the CSRG, overseeing the development of over 400 freely redistributable UNIX-compatible utilities, and is the recipient of the 1991 Distinguished Achievement Award from the University of California, Berkeley, for his work to make 4.4BSD freely redistributable. Concurrently, he was the principal architect of the 2.10BSD release of the Berkeley Software Distribution for PDP-11s, and the coauthor of the Berkeley Log Structured Filesystem and the Berkeley database package (*DB*). He is also the author of the widely used *vi* implementation, *nvi*. He received his undergraduate degree in Statistics and his Masters degree in Electrical Engineering from George Washington University. He is a member of the ACM, the IEEE, and several POSIX working groups. In his spare time, he enjoys scuba diving in the South Pacific, mountain biking, and working on a tunnel into Kirk and Eric's specially constructed wine cellar. He lives in Massachusetts with his wife, Margo Seltzer, and their cats.

Michael J. Karels is the System Architect and Vice President of Engineering at Berkeley Software Design, Inc. He spent 8 years as the Principal Programmer of the CSRG at the University of California, Berkeley as the system architect for 4.3BSD. Karels received his Bachelor's degree in Microbiology from the University of Notre Dame. While a graduate student in Molecular Biology at the University of California, he was the principal developer of the 2.9BSD UNIX release of the Berkeley Software Distribution for the PDP-11. He is a member of the ACM, the IEEE, and several POSIX working groups. He lives with his wife Teri Karels in the backwoods of Minnesota.

John S. Quarterman is a partner in Texas Internet Consulting (TIC), which consults in networks and open systems with particular emphasis on TCP/IP networks, UNIX systems, and standards. He is the author of *The Matrix: Computer Networks and Conferencing Systems Worldwide* (Digital Press, 1990), and is a coauthor of *UNIX, POSIX, and Open Systems: The Open Standards Puzzle* (1993), *Practical Internetworking with TCP/IP and UNIX* (1993), *The Internet Connection: System Connectivity and Configuration* (1994), and *The E-Mail Companion: Communicating Effectively via the Internet and Other Global Networks* (1994), all published by Addison-Wesley. He is editor of *Matrix News*, a monthly newsletter about issues that cross network, geographic, and political boundaries, and of *Matrix Maps Quarterly*; both are published by Matrix Information and Directory Services, Inc. (MIDS) of Austin, Texas. He is a partner in Zilker Internet Park, which provides Internet access from Austin. He and his wife, Gretchen Quarterman, split their time among his home in Austin, hers in Buffalo, New York, and various other locations.

Contents

Part 1 Overview	1
Chapter 1 History and Goals	3
1.1 History of the UNIX System	3
Origins	3
Research UNIX	4
AT&T UNIX System III and System V	7
Other Organizations	8
Berkeley Software Distributions	8
UNIX in the World	10
1.2 BSD and Other Systems	10
The Influence of the User Community	11
1.3 Design Goals of 4BSD	12
4.2BSD Design Goals	13
4.3BSD Design Goals	14
4.4BSD Design Goals	15
1.4 Release Engineering	16
References	17
Chapter 2 Design Overview of 4.4BSD	21
2.1 4.4BSD Facilities and the Kernel	21
The Kernel	22
2.2 Kernel Organization	23
2.3 Kernel Services	25
2.4 Process Management	26
Signals	27
Process Groups and Sessions	28
2.5 Memory Management	29
BSD Memory-Management Design Decisions	29