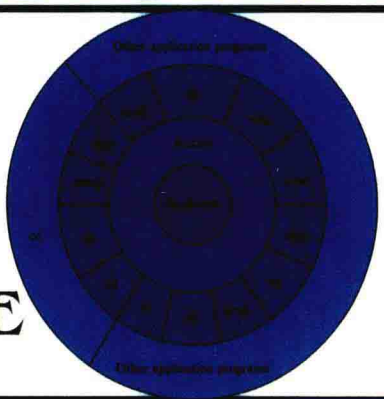
HZ BOOKS 经典原版书库

UNIX操作系统设计

（英文版）

THE DESIGN OF THE UNIX™ OPERATING SYSTEM



MAURICE J. BACH

PRENTICE HALL SOFTWARE

（美）Maurice J. Bach 著

机械工业出版社
China Machine Press

UNIX操作系统设计 (英文版)

The Design of the UNIX Operating System

Linux之父Linus Torvalds曾捧读的经典著作

本书是一本全面介绍UNIX系统V内核结构的经典教材。Bach在这本传世之作中深入分析了UNIX的内核算法、基本数据结构以及它们同上层编程接口的关系。本书首先对系统内核结构进行了简要介绍，然后分章节描述了文件系统、进程调度和存储管理，并在此基础上讨论了UNIX系统的高级问题，如驱动程序接口、进程间通信与网络等。

本书虽然以UNIX系统V为背景，但是介绍的算法、数据结构却并没有专门针对任何一种特定的内核，所以直到今日，本书仍然是世界上许多大学操作系统课程的必读或推荐教材。读者如果想要学习UNIX，本书依然是最好的选择之一。

本书的适用范围非常广泛。首先，本书可用作高等院校高年级本科生或低年级研究生的操作系统课程教材，学生使用本书的同时若能参考系统源代码将获益匪浅，但也可以独立地学习本书。其次，系统程序员可将本书作为参考书，从而更好地理解内核的工作原理，并将UNIX系统中采用的算法与其他操作系统的算法加以比较。最后，UNIX系统程序员也可将本书作为参考书，从而更深入地了解他们的程序是如何与系统相互作用的，进而编写出更有效、更高级的程序。

(中文版)
ISBN 7-111-07850-0
定价：35.00元

www.PearsonEd.com

上架指导：计算机/操作系统

ISBN 7-111-19765-8

9 787111 197652

封面设计：吴 刚

华章网站 http://www.hzbook.com

网上购书：www.china-pub.com

投稿热线：(010) 88379604

购书热线：(010) 68995259, 68995264

读者信箱：hzjsj@hzbook.com

ISBN 7-111-19765-8

定价：49.00元

HZ BOOKS

UNIX操作系统设计

The Design of the UNIX Operating System

(美) Maurice J. Bach 著

英文版

版社
Press

经 典 原 版 书 库

UNIX操作系统设计

（英文版）

The Design of the UNIX Operating System

（美）Maurice J. Bach 著

机械工业出版社
China Machine Press

本书版权登记号：图字：01-2006-3996

图书在版编目（CIP）数据

UNIX操作系统设计（英文版）/（美）巴赫（Bach, M. J.）著. -北京：机械工业出版社，2006.9

（经典原版书库）

书名原文：The Design of the UNIX Operating System

ISBN 7-111-19765-8

I. U… II. 巴… III. UNIX操作系统-程序设计-英文 IV. TP316. 81

中国版本图书馆CIP数据核字（2006）第096645号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：迟振春

三河市明辉印装有限公司印刷 · 新华书店北京发行所发行

2006年9月第1版第1次印刷

170mm × 242mm · 30.5印张

定价：49.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书热线：（010）68326294

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall，Addison-Wesley，McGraw-Hill，Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum，Stroustrup，Kernighan，Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及庋藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也

为了更好地为学校和老师们服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业的教学度身订造的。其中许多教材均已为M. I. T.，Stanford，U.C. Berkeley，C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

电子邮件：hzjsj@hzbook.com
联系电话：（010）68995264
联系地址：北京市西城区百万庄南街1号
邮政编码：100037

专家指导委员会

（按姓氏笔画顺序）

尤晋元	王　珊	冯博琴	史忠植	史美林
石教英	吕　建	孙玉芳	吴世忠	吴时霖
张立昂	李伟琴	李师贤	李建中	杨冬青
邵维忠	陆丽娜	陆鑫达	陈向群	周伯生
周克定	周傲英	孟小峰	岳丽华	范　明
郑国梁	施伯乐	钟玉琢	唐世渭	袁崇义
高传善	梅　宏	程　旭	程时端	谢希仁
裘宗燕	戴　葵			

To my parents, for their patience and devotion,
to my daughters, Sarah and Rachel, for their laughter,
to my son, Joseph, who arrived after the first printing,
and to my wife, Debby, for her love and understanding.

PREFACE

The UNIX system was first described in a 1974 paper in the Communications of the ACM [Thompson 74] by Ken Thompson and Dennis Ritchie. Since that time, it has become increasingly widespread and popular throughout the computer industry where more and more vendors are offering support for it on their machines. It is especially popular in universities where it is frequently used for operating systems research and case studies.

Many books and papers have described parts of the system, among them, two special issues of the Bell System Technical Journal in 1978 [BSTJ 78] and 1984 [BLTJ 84]. Many books describe the user level interface, particularly how to use electronic mail, how to prepare documents, or how to use the command interpreter called the shell; some books such as *The UNIX Programming Environment* [Kernighan 84] and *Advanced UNIX Programming* [Rochkind 85] describe the programming interface. This book describes the internal algorithms and structures that form the basis of the operating system (called the kernel) and their relationship to the programmer interface. It is thus applicable to several environments. First, it can be used as a textbook for an operating systems course at either the advanced undergraduate or first-year graduate level. It is most beneficial to reference the system source code when using the book, but the book can be read independently, too. Second, system programmers can use the book as a reference to gain better understanding of how the kernel works and to compare algorithms used in the UNIX system to algorithms used in other operating systems.

Finally, programmers on UNIX systems can gain a deeper understanding of how their programs interact with the system and thereby code more-efficient, sophisticated programs.

The material and organization for the book grew out of a course that I prepared and taught at AT&T Bell Laboratories during 1983 and 1984. While the course centered on reading the source code for the system, I found that understanding the code was easier once the concepts of the algorithms had been mastered. I have attempted to keep the descriptions of algorithms in this book as simple as possible, reflecting in a small way the simplicity and elegance of the system it describes. Thus, the book is not a line-by-line rendition of the system written in English; it is a description of the general flow of the various algorithms, and most important, a description of how they interact with each other. Algorithms are presented in a C-like pseudo-code to aid the reader in understanding the natural language description, and their names correspond to the procedure names in the kernel. Figures depict the relationship between various data structures as the system manipulates them. In later chapters, small C programs illustrate many system concepts as they manifest themselves to users. In the interests of space and clarity, these examples do not usually check for error conditions, something that should always be done when writing programs. I have run them on System V; except for programs that exercise features specific to System V, they should run on other versions of the system, too.

Many exercises originally prepared for the course have been included at the end of each chapter, and they are a key part of the book. Some exercises are straightforward, designed to illustrate concepts brought out in the text. Others are more difficult, designed to help the reader understand the system at a deeper level. Finally, some are exploratory in nature, designed for investigation as a research problem. Difficult exercises are marked with asterisks.

The system description is based on UNIX System V Release 2 supported by AT&T, with some new features from Release 3. This is the system with which I am most familiar, but I have tried to portray interesting contributions of other variations to the operating system, particularly those of Berkeley Software Distribution (BSD). I have avoided issues that assume particular hardware characteristics, trying to cover the kernel-hardware interface in general terms and ignoring particular machine idiosyncrasies. Where machine-specific issues are important to understand implementation of the kernel, however, I delve into the relevant detail. At the very least, examination of these topics will highlight the parts of the operating system that are the most machine dependent.

The reader must have programming experience with a high-level language and, preferably, with an assembly language as a prerequisite for understanding this book. It is recommended that the reader have experience working with the UNIX system and that the reader knows the C language [Kernighan 78]. However, I have attempted to write this book in such a way that the reader should still be able to absorb the material without such background. The appendix contains a simplified description of the system calls, sufficient to understand the presentation

in the book, but not a complete reference manual.

The book is organized as follows. Chapter 1 is the introduction, giving a brief, general description of system features as perceived by the user and describing the system structure. Chapter 2 describes the general outline of the kernel architecture and presents some basic concepts. The remainder of the book follows the outline presented by the system architecture, describing the various components in a building block fashion. It can be divided into three parts: the file system, process control, and advanced topics. The file system is presented first, because its concepts are easier than those for process control. Thus, Chapter 3 describes the system buffer cache mechanism that is the foundation of the file system. Chapter 4 describes the data structures and algorithms used internally by the file system. These algorithms use the algorithms explained in Chapter 3 and take care of the internal bookkeeping needed for managing user files. Chapter 5 describes the system calls that provide the user interface to the file system; they use the algorithms in Chapter 4 to access user files.

Chapter 6 turns to the control of processes. It defines the context of a process and investigates the internal kernel primitives that manipulate the process context. In particular, it considers the system call interface, interrupt handling, and the context switch. Chapter 7 presents the system calls that control the process context. Chapter 8 deals with process scheduling, and Chapter 9 covers memory management, including swapping and paging systems.

Chapter 10 outlines general driver interfaces, with specific discussion of disk drivers and terminal drivers. Although devices are logically part of the file system, their discussion is deferred until here because of issues in process control that arise in terminal drivers. This chapter also acts as a bridge to the more advanced topics presented in the rest of the book. Chapter 11 covers interprocess communication and networking, including System V messages, shared memory and semaphores, and BSD sockets. Chapter 12 explains tightly coupled multiprocessor UNIX systems, and Chapter 13 investigates loosely coupled distributed systems.

The material in the first nine chapters could be covered in a one-semester course on operating systems, and the material in the remaining chapters could be covered in advanced seminars with various projects being done in parallel.

A few caveats must be made at this time. No attempt has been made to describe system performance in absolute terms, nor is there any attempt to suggest configuration parameters for a system installation. Such data is likely to vary according to machine type, hardware configuration, system version and implementation, and application mix. Similarly, I have made a conscious effort to avoid predicting future development of UNIX operating system features. Discussion of advanced topics does not imply a commitment by AT&T to provide particular features, nor should it even imply that particular areas are under investigation.

It is my pleasure to acknowledge the assistance of many friends and colleagues who encouraged me while I wrote this book and provided constructive criticism of the manuscript. My deepest appreciation goes to Ian Johnstone, who suggested

that I write this book, gave me early encouragement, and reviewed the earliest draft of the first chapters. Ian taught me many tricks of the trade, and I will always be indebted to him. Doris Ryan also had a hand in encouraging me from the very beginning, and I will always appreciate her kindness and thoughtfulness. Dennis Ritchie freely answered numerous questions on the historical and technical background of the system. Many people gave freely of their time and energy to review drafts of the manuscript, and this book owes a lot to their detailed comments. They are Debby Bach, Doug Bayer, Lenny Brandwein, Steve Buroff, Tom Butler, Ron Gomes, Mesut Gunduc, Laura Israel, Dean Jagels, Keith Kelleman, Brian Kernighan, Bob Martin, Bob Mitze, Dave Nowitz, Michael Poppers, Marilyn Safran, Curt Schimmel, Zvi Spitz, Tom Vaden, Bill Weber, Larry Wehr, and Bob Zarrow. Mary Fruhstuck provided help in preparing the manuscript for typesetting. I would like to thank my management for their continued support throughout this project and my colleagues, for providing such a stimulating atmosphere and wonderful work environment at AT&T Bell Laboratories. John Wait and the staff at Prentice-Hall provided much valuable assitance and advice to get the book into its final form. Last, but not least, my wife, Debby, gave me lots of emotional support, without which I could never have succeeded.

CONTENTS