附光盘

大学计算机教育丛书（影印版）

Second Edition

OPERATING SYSTEMS

Design and Implementation

操作系统

设计及实现

（第二版）

Andrew S. Tanenbaum
Albert S. Woodhull

清华大学出版社 · PRENTICE HALL

OPERATING SYSTEMS:

Design and Implementation
Second Edition

操 作 系 统
设 计 及 实 现
（第二版）

Andrew S. Tanenbaum
Albert S. Woodhull

清 华 大 学 出 版 社
Prentice-Hall International Inc.

(京) 新登字 158 号

Operating Systems: design and implementation 2nd Ed. /Andrew S. Tanenbaum, Albert S. Woodhull

图书在版编目(CIP)数据
操作系统: 设计及实现:第二版:英文/(美)特南鲍姆(Tanenbaum A.S),(美)伍德哈尔(Woodhull A.S)著 .-影印本 .-北京:清华大学出版社,1997.9
(大学计算机教育丛书)
ISBN 7-302-02714-5

Ⅰ.操… Ⅱ.①特… ②伍… Ⅲ.操作系统(软件) Ⅳ.TP316

中国版本图书馆 CIP 数据核字(97)第 17357 号

出版者: 清华大学出版社(北京清华大学校内,邮编 100084)
　　　　因特网地址: www.tup.tsinghua.edu.cn
印刷者: 清华大学印刷厂
发行者: 新华书店总店北京科技发行所
开　本: 850×1168 1/32　印张: 30
版　次: 1997 年 9 月第 1 版　1998 年 7 月第 3 次印刷
书　号: ISBN 7-302-02714-5/TP・1405
印　数: 10001～15000
定　价: 69.00 元

出 版 前 言

我们的大学生、研究生毕业后，面临的将是一个国际化的信息时代。他们将需要随时查阅大量的外文资料；会有更多的机会参加国际性学术交流活动；接待外国学者；走上国际会议的讲坛。作为科技工作者，他们不仅应有与国外同行进行口头和书面交流的能力，更为重要的是，他们必须具备极强的查阅外文资料获取信息的能力。有鉴于此，在国家教委所颁布的“大学英语教学大纲”中有一条规定：专业阅读应作为必修课程开设。同时，在大纲中还规定了这门课程的学时和教学要求。有些高校除开设“专业阅读”课之外，还在某些专业课拟进行英语授课。但教、学双方都苦于没有一定数量的合适的英文原版教材作为教学参考书。为满足这方面的需要，我们挑选了 7 本计算机科学方面最新版本的教材，进行影印出版。Prentice Hall 公司和清华大学出版社这次合作将国际先进水平的教材引入我国高等学校，为师生们提供了教学用书，相信会对高校教材改革产生积极的影响。

清华大学出版社
Prentice Hall 公司

1996. 11

OPERATING SYSTEMS:

DESIGN AND IMPLEMENTATION

Second Edition

PREFACE

Most books on operating systems are strong on theory and weak on practice. This one aims to provide a better balance between the two. It covers all the fundamental principles in great detail, including processes, interprocess communication, semaphores, monitors, message passing, scheduling algorithms, input/output, deadlocks, device drivers, memory management, paging algorithms, file system design, security, and protection mechanisms. But it also discusses one particular system—MINIX, a UNIX-compatible operating system—in detail, and even provides a complete source code listing for study. This arrangement allows the reader not only to learn the principles, but also to see how they are applied in a real operating system.

When the first edition of this book appeared in 1987, it caused something of a small revolution in the way operating systems courses were taught. Until then, most courses just covered theory. With the appearance of MINIX, many schools began to have laboratory courses in which students examined a real operating system to see how it worked inside. We consider this trend highly desirable and hope this second edition strengthens it.

It its first 10 years, MINIX has undergone many changes. The original code was designed for a 256K 8088-based IBM PC with two diskette drives and no hard disk. It was also based on Version 7 of UNIX. As time went on, MINIX evolved in many ways. For example, the current version will now run on anything from the original PC (in 16-bit real mode) to large Pentiums with massive hard disks (in 32-bit protected mode). It also changed from being based on Version 7, to being based on the international POSIX standard (IEEE 1003.1 and ISO 9945-1) Finally, many features were added, perhaps too many in our view, but too few in the view of some other people, which led to the creation of LINUX. In addition, MINIX was ported to many other platforms, including the Macintosh, Amiga, Atari, and SPARC. This book covers only MINIX 2.0, which so far runs only on computers with an 80x86 CPU, on systems which can emulate such a CPU, or on the SPARC.

This second edition of the book has many changes throughout. Nearly all of the material on principles has been revised, and considerable new material has been added. However, the main change is the discussion of the new, POSIX-based MINIX, and the inclusion of the new code in this book. Also new is the inclusion of a CD-ROM in each book containing the full MINIX source code plus instructions for installing MINIX on a PC (see the file README.TXT in the main CD-ROM directory).

Setting up MINIX on an 80x86 PC, whether for individual use or for a laboratory is straightforward. A disk partition of at least 30 MB must be made for it, then it can be installed by just following the instructions in the *README.TXT* file on the CD-ROM. To print the *README.TXT* file on a PC, first start MS-DOS, if it is not already running (from WINDOWS, click on the MS-DOS) icon. Then type

```
copy readme.txt prn
```

to make the printout. The file can also be examined in *edit*, *wordpad*, *notepad*, or any other text editor that can handle flat ASCII text.

For schools (or individuals) that do not have PCs available, two other options are now available. Two simulators are included on the CD-ROM. One, written by Paul Ashton, runs on SPARCs. It runs MINIX as a user program on top of Solaris. As a consequence, MINIX is compiled into a SPARC binary and runs at full speed. In this mode, MINIX is no longer an operating system, but a user program, so some changes to the low-level code were necessary.

The other simulator was written by Kevin P. Lawton of Bochs Software Company. This simulator interprets the Intel 80386 instruction set and enough I/O gear that MINIX can run on the simulator. Of course running on top of an interpreter costs some performance, but it makes debugging much easier for students. This simulator has the advantage that it will run on any computer that supports the M.I.T. X Window System. For more information about both simulators, please see the CD-ROM.

The development of MINIX is an ongoing proposition. The contents of this book and its CD-ROM are merely a snapshot of the system as of the time of publication. For the current state of affairs, please see the MINIX home page on the World Wide Wide, *http://www.cs.vu.nl/~ast/minix.html*. In addition, MINIX has its own USENET newsgroup: *comp.os.minix*, to which readers can subscribe to find out what is going on in the MINIX world. For those with e-mail, but without newsgroup access, there is also a mailing list. Write to *listserv@listserv.nodak.edu* with "subscribe minix-l <your full name>" as the first and only line in the body of the message. You will receive more information by return e-mail.

For classroom use, a problem solutions manual is available, to instructors only, from Prentice Hall. PostScript files containing all the figures in the book, suitable for making overhead sheets, can be found by following the link marked "Software and supplementary material" from *http://www.cs.vu.nl/~ast/*.

We have been extremely fortunate in having the help of many people during the course of this project. First and foremost, we would like to thank Kees Bot for doing the lion's share of the work in making MINIX conform to the standard and for managing the distribution. Without his enormous help, we would never have made it. He wrote large chunks of code himself (e.g. the POSIX terminal I/O), cleaned up other sections, and repaired numerous bugs that had crept in over the years. Thank you for a job well done.

Bruce Evans, Philip Homburg, Will Rose, and Michael Temari have all contributed to the development of MINIX over the years. Hundreds of other people have contributed to MINIX via the newsgroup. There were so many of them and their contributions have been so varied that we cannot even begin to list them all here, so the best we can do is a generic thank you to all of them.

Several people read parts of the manuscript and made suggestions. We would like to give our special thanks to John Casey, Dale Grit, and Frans Kaashoek.

A number of students at the Vrije Universiteit tested the beta version of the CD-ROM. These were: Ahmed Batou, Goran Dokic, Peter Gijzel, Thomer Gil, Dennis Grimbergen, Roderick Groesbeek, Wouter Haring, Guido Kollerie, Mark Lassche, Raymond Ris, Frans ter Borg, Alex van Ballegooy, Ries van der Velden, Alexander Wels, and Thomas Zeeman. We would like to thank all of them for their careful work and detailed reports.

ASW would also like to thank several of his former students, particularly Peter W. Young of Hampshire College and Maria Isabel Sanchez and William Puddy Vargas of the Universidad Nacional Autonoma de Nicaragua for the part their interest in MINIX played in sustaining his efforts.

Finally, we would like to thank our families. Suzanne has been through this ten times now. Barbara has been through it nine times now. Marvin has been through it eight times now. Even Little Bram has been through it four times. It's kind of getting to be routine, but the love and support is still much appreciated. (ast)

As for Al's Barbara, this is the first time she has been through this. It would not have been possible without her support, patience, and good humor. It has been Gordon's good fortune to have been away at college through most of this. But it is a delight to have a son who understands and cares about the same things that fascinate me. (asw)

Andrew S. Tanenbaum
Albert S. Woodhull

ABOUT THE AUTHORS

Andrew S. Tanenbaum has an S.B. degree from M.I.T. and a Ph.D. from the University of California at Berkeley. He is currently a Professor of Computer Science at the Vrije Universiteit in Amsterdam, The Netherlands, where he heads the Computer Systems Group. He is also Dean of the Advanced School for Computing and Imaging, an inter-university graduate school doing research on advanced parallel, distributed, and imaging systems. Nevertheless, he is trying very hard to avoid turning into a bureaucrat.

In the past, he has done research on compilers, operating systems, networking, and local-area distributed systems. His current research focuses primarily on the design of wide-area distributed systems that scale to millions of users. These research projects have led to over 70 refereed papers in journals and conference proceedings, and five books.

Prof. Tanenbaum has also produced a considerable volume of software. He was the principal architect of the Amsterdam Compiler Kit, a widely-used toolkit for writing portable compilers, as well as of MINIX. Together with his Ph.D. students and programmers, he helped design the Amoeba distributed operating system, a high-performance microkernel-based distributed operating system. MINIX and Amoeba are now available for free for education and research via the Internet.

His Ph.D. students have gone on to greater glory after getting their degrees. He is very proud of them. In this respect he resembles a mother hen.

Prof. Tanenbaum is a Fellow of the ACM, a Senior Member of the IEEE, a member of the Royal Netherlands Academy of Arts and Sciences, winner of the 1994 ACM Karl V. Karlstrom Outstanding Educator Award, and winner of the 1997 ACM/SIGCSE Award for Outstanding Contributions to Computer Science Education. He is also listed in *Who's Who in the World.* His home page on the World Wide Web can be found at URL *http://www.cs.vu.nl/~ast/* .

Albert S. Woodhull has an S.B. degree from M.I.T. and a Ph.D. from the University of Washington. He entered M.I.T. intending to become an electrical engineer, but he emerged as a biologist. He has been associated with the School of Natural Science of Hampshire College in Amherst, Massachusetts, since 1973. As a biologist using electronic instrumentation, he started working with microcomputers when they became readily available. His instrumentation courses for science students evolved into courses in computer interfacing and real-time programming.

Dr. Woodhull has always had strong interests in teaching and in the role of science and technology in development. Before entering graduate school he taught high school science for two years in Nigeria. More recently he spent several sabbaticals teaching computer science at Nicaragua's Universidad Nacional de Ingenieria and the Universidad Nacional Autonoma de Nicaragua.

He is interested in computers as electronic systems, and in interactions of computers with other electronic systems. He particularly enjoys teaching in the areas of computer architecture, assembly language programming, operating systems, and computer communications. He has also worked as a consultant in the development of electronic instrumentation and related software.

He has many nonacademic interests as well, including various outdoor sports, amateur radio, and reading. He enjoys travelling and trying to make himself understood in languages other than his native English. His World Wide Web home page is located on a system running MINIX, at URL *http://minix1.hampshire.edu/asw/* .

CONTENTS

APPENDICES