国外计算机科学教材系列

操作系统:设计与实现

(第2版)

(上册)

OPERATING SYSTEMS

Design and Implementation (Second Edition)

ANDREW S. TANENBAUM
ALBERT S. WOODHULL 著

王 鹏 尤晋元
朱 鹏 敖青云 译校

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

PRENTICE HALL 出版公司

国外计算机科学教材系列

操作系统：设计与实现

（第2版）

（上册）

Operating Systems

Design and Implementation

second edition

Andrew S. Tanenbaum, Albert S. Woodhull 著

王 鹏 尤晋元 朱 鹏 敖青云 译校

PRENTICE HALL出版公司

電子工業出版社

内 容 提 要

本书共6章，涵盖了操作系统课程的所有内容。即传统上的进程管理、存储器管理、文件管理和设备管理，同时又包含线程、基于消息传递的系统的构造模型、日志结构文件系统、安全保护机制、RAM及CD-ROM盘等，且以Pentium CPU作为实例。这样，既能学习操作系统的经典内容，又能了解当前最新技术。

本书为第二版，其第一版于1987年出版时，曾引发了操作系统课程教学的一场小变革。因为，在那以前多数教材只讲理论，而本教材却是基于理论与具体实例(MINIX)的结合。这对于掌握操作系统的设计与实现是大有裨益的。

本书分为上、下两册。上册为正文部分；下册为三个附录及随书光盘。

丛 书 名：国外计算机科学教材系列
原 书 名：OPERATING SYSTEMS: Design and Implementation (second edition)
书　　名：**操作系统：设计与实现（第2版）**
（上册）
著　　者：Andrew S. Tanenbaum, Albert S. Woodhull 著
译 校 者：王 鹏　尤晋元　朱 鹏　敖青云
责任编辑：邓又强
印 刷 者：顺义县天竺颖华印刷厂印刷
出版发行：电子工业出版社出版、发行　URL: http://www.phei.com.cn
北京市海淀区万寿路173信箱　邮编　100036　发行部电话 68214070
经　　销：各地新华书店经销
开　　本：787×1092 1/16　印张：25.5　字数：593千字
版　　次：1998年8月第1版　1998年8月第1次印刷
印　　数：7000册
书　　号：ISBN 7-5053-4774-8/TP·2310
定　　价：40.00元
著作权合同登记号　图字：01-98-0965

出版说明

计算机科学的迅速发展是20世纪科学发展史上最伟大的事件之一。从1946年第一台笨重而体积庞大的计算机的发明至今，仅仅半个多世纪，计算机已经变得小巧无比却又能力非凡。它的应用已经渗透到了社会的各个方面，成为当今所谓的信息社会的最显著的特征。

处于世纪之交科技进步的大潮中，我国正在加强计算机科学的高等教育，着眼于为下一世纪培养高素质的计算机人才，以适应信息社会加速度发展的需要。当前，全国各类高等院校已经或计划在各专业基础课程规划中增加计算机科学的课程内容，而作为与计算机科学密切相关的计算机、通信、信息等专业，更是在酝酿着教学的全面革新，以期规划出一整套面向21世纪的、具有中国高校计算机教育特色的课程计划和教材体系。值此，我们不妨借鉴并引进国外具有先进性、实用性和权威性的大学计算机教材，洋为中用，以更好地服务于国内的高校教育。

美国Prentice Hall出版公司是享誉世界的高校教材出版商，自1913年公司成立以来，即致力于教育图书的出版。它所出版的计算机教材在美国为众多大学所采用，其中有不少是专业领域中的经典名著。许多蜚声世界的教授学者成为该公司的资深作者，如：道格拉斯·科默(Douglas Comer)，安德鲁·坦尼伯姆(Andrew Tanenbaum)，威廉·斯大林(William Stallings)……几十年来，他们的著作教育了一批批不同肤色的莘莘学子，使这些教材同时也成为全人类的共同财富。

为了保证本系列教材翻译出版的质量，电子工业出版社和Prentice Hall出版公司共同约请北京地区的清华大学、北京大学、北京航空航天大学，上海地区的上海交通大学、复旦大学，南京地区的南京大学、解放军通信工程学院等全国著名的高等院校的教学第一线的几十位教师参加翻译工作。这中间有正在讲授同类教材的年轻教师和博士，有积累了几十年教学经验的教授和博士生导师，还有我国著名的计算机科学家。他们的辛勤劳动保证了本系列丛书得以高质量地出版面世。

如此大规模地引进计算机科学系列教材，在我们还是第一次。除缺乏经验之外，还由于我们对计算机科学的发展，对中国高校计算机教育特点认识的不足，致使在选题确定、翻译、出版等工作中，肯定存在许多遗憾和不足之处，恳请广大师生和其他读者提出批评、建议。

电子工业出版社
URL：http://www.phei.com.cn
Prentice Hall出版公司
URL：http://www.prenhall.com

译者序

坦尼鲍姆教授是国际知名的计算机科学家和教育家。他在操作系统、分布式系统以及计算机网络领域都有很深的造诣。自80年代以来，他已先后出版了一系列面向大学生和研究生的教材性质的专著，并被世界各国的许多大学广泛采用。这本书就是他的最新专著之一。

操作系统是计算机系统中最核心和最底层的软件，对操作系统的深入学习关系到对整个系统运作机制的全面理解，因此一本好教材也显得愈发重要。本书的英文版出版于1997年，其中涵盖了操作系统课程的所有内容，即传统上的进程管理、存储器管理、文件管理和设备管理。同时其中又包含了许多新内容，如线程、基于消息传递的系统构造模型、日志结构文件系统、安全和保护机制、RAM盘及CD-ROM设备等，而用作例子的CPU则为Interl Pentium。这使得读者一方面能够学习操作系统的经典内容，另一方面又能够了解和跟踪当前的最新技术和研究成果。

本书的另一个特点是基本原理与具体实例，即MINIX紧密结合。第2到第5章的前半部分讲述原理，后半部分则详细地解释这些原理在MINIX的设计和实现中的应用。通过阅读这些部分能够把握MINIX源代码的组织方式，并理解那些很关键或者很难懂的代码。这部分内容非常翔实，有时甚至逐行地解释附录中所列的源程序。对操作系统课程多年的授课经验以及相关的科研工作使我们认识到：详细地剖析一个像MINIX这样的操作系统对于掌握操作系统设计与实现的精髓是大有裨益的。

正因为上述原因，我们真切地感受到将这本书翻译、介绍给国内读者将是一件非常有意义的事，衷心希望我们付出的劳动能对国内的操作系统教学和实践有所帮助和促进。

本书的第1章，第2章，第3章由王鹏翻译，刘福岩和陆宁也参加了部分工作；第四章由朱鹏翻译；第五章由敖青云翻译。全书由尤晋元教授审校并统稿。

在整个翻译过程中，上海交通大学计算机系系统软件研究室的师生给予了许多帮助。并且在计算机系95级本科生的操作系统课程中进行了试用，许多学生提出了很好的建议，在此向他们表示衷心的感谢。

特别要感谢本书的责任编辑邓又强编审，本书的顺利出版与他的辛勤劳动和热情支持是分不开的。

虽然在翻译过程中我们尽力恪守“信，达，雅”的准则，但不当和疏漏之处在所难免，敬请读者提出宝贵建议。

译　者

上海交通大学计算机科学与工程系

1998.4

前　言

多数操作系统教材都重理论而轻实践，本书希望在这二者之间求取较好的平衡。本书详细论述了操作系统的所有基本概念，包括进程、进程间通信、信号量、管程、消息传递、调度算法、输入/输出、死锁、设备驱动程序、存储器管理、页面调度算法、文件系统设计、安全与保护机制等。同时，本书也详细讨论了 MINIX——一个与 UNIX 兼容的操作系统，并提供了完整的源代码供学习之用。这样的安排使读者不仅学习到理论，而且能够理解它们如何应用在一个实际的操作系统之中。

本书第一版在 1987 年出版时，曾引发了操作系统课程教学的一场小小的变革。在此之前多数课程都只讲理论。随着 MINIX 的出现，许多学校开始增加实验环节以使学生了解实际的操作系统是如何运作的。我们认为这种趋势是可取的，并希望通过本书第二版能进一步加强这种趋势。

MINIX 在其出现以来的十年间发生了许多变化，最初的代码是为基于 8088 芯片、256K 内存和两个软驱的 IBM PC 机型编写的，它基于 UNIX 版本 7。随着时间的推移，MINIX 在许多方面有所发展，比如当前版本可运行在众多机型上，从 16 位实模式的 PC 机到配有大容量硬盘的奔腾机(32 位保护模式)，而且它不再基于 UNIX 版本 7，而是基于国际上的 POSIX 标准(POSIX 1003.1 和 ISO9945-1)。与此同时，有许多新特征被添加到 MINIX 中，在我们看来，所增加的特征可能已经太多了，但有些人则认为还不够，这最终导致了 LINUX 的诞生。MINIX 还被移植到许多其他平台上，包括 Macintosh、Amiga、Atari 和 SPARC。本书只涉及 MINIX2.0，到目前为止，该版本只能运行于基于 80x86 的机器，或者可模拟此类 CPU 的机器，以及 SPARC 机器。

与第一版相比，第二版有许多变化，原理性部分基本都被修改过，同时增加了大量新内容。最主要的变化是新的基于 POSIX 的 MINIX，以及对其源代码的剖析。另外，每本书都附带一张 CD－ROM，它包含了全部 MINIX 源代码，以及在 PC 上安装 MINIX 的说明(见 CD－ROM 主目录下的 README.TXT 文件)。

在一台 80x86 的 PC 机上安装 MINIX 很方便。它需要一个至少 30MB 的硬盘分区，然后按照 CD-ROM 上 README.TXT 文件中的步骤进行即可。在打印 README.TXT 文件之前，先启动 MS－DOS(若运行 WINDOWS，则双击 MS-DOS 图标)，然后键入

```
copy readme.txt prn
```

即可。该文件也可以用 edit、wordpad、notepad 等任何可以处理 ASCII 正文的编辑器进行浏览。

对于没有 PC 机的学校和个人，有两种解决办法，即 CD-ROM 上提供的两个模拟程序。一个由 Paul Ashton 为 SPARC 机器编写，它作为用户程序在 Solaris 上运行，此时 MINIX 被编译成 SPARC 上的可执行文件。在这种模式下，MINIX 不再是一个操作系统，而只是一个用户程序，所以必须对其底层作一些修改。

另一个模拟程序由 Bochs 软件公司的 Kevin P. Lawton 编写，它解释 Intel 80386 的指令集以及足以使 MINIX 运行所需的 I/O 指令。显然，在解释器层次上运行使性能有所下降，但这使得学生更容易进行调试。该模拟程序运行在所有支持 M. I. T 的 X-Window 的系统上，更详细的信息请参看 CD-ROM 上的有关文件。

MINIX 仍在继续发展，本书和 CD -ROM 中的内容仅仅反映了本书出版时的情况，有关 MINIX 的最新动态请访问 MINIX 的主页：http://www.cs.vu.nl/~ast/minix.html。MINIX 也有 USENET 中的新闻组：comp.os.minix，读者可以订阅该新闻组。对于仅有 Email 的读者可通过以下步骤来加入 MINIX 的邮件用户通信组。给 listserv@listserv.nodak.edu 发一封信，其中只需一行字："subscribeminix-1＜您的完整用户名＞"，此后你便会通过 E-mail 获得很多的信息。

讲授本课程的教师可以从 Prentice Hall 出版公司获得一份习题解答手册。从 WWW 地址 http://www.cs.vu.nl/ast/沿着"Software and supplementary material"链接可以获得一些有用的 PostScript 文件，其中包含本书中所有的图表，可供需要时使用。

在 MINIX 的开发项目中我们有幸得到了许多人的帮助。首先要感谢 Kees Bot 在 MINIX 标准化和软件发布中所作的大量工作，没有他的帮助，我们不可能完成这件工作。他自己编写了大量的代码(如 POSIX 终端 I/O)并修正了一些数年来一直存在的错误，他还整理了其他的代码。

这些年来 Bruce Evans, Philip Homburg, Will Rose 和 Michael Temari 为 MINIX 的开发做了大量的工作。有几百人通过新闻组对 MINIX 作出了贡献，他们人数众多，所作出的贡献也各不相同，在此谨向他们一并表示感谢。

John Casey, Dale Grit, Frans Kaashoek 等人阅读了本书的部分手稿并提出了宝贵建议，在此向他们表示谢意。

Vrije 大学的许多学生测试了 CD-ROM 中 MINIX 的 β 版本，他们是：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 以及 Thomas Zeeman。我们对他们细致的工作和详尽的报告致以衷心的感谢。

阿尔伯特·S·伍德豪尔向他从前的几位学生表示感谢，特别是 Hampshire 学院的 Peter W. Young，Nacional Autonoma de Nicaragua 大学的 Maria Isabel Sanchez 和 William Puddy Vargas。

最后要向我们的家庭成员表示感谢。Suzanne 已是第十次在我埋头写作时给我支持，对 Barbara 是第九次，Marvin 是第八次，甚至小 Bram 也是第四次了。他们的支持和爱心对我非常重要。(坦尼鲍姆)

至于阿尔伯特的 Barbara，这倒是第一次，假如没有她的支持，耐心和幽默，我们是不可能完成这一工作的，对我的儿子 Gordon 而言，由于在编写本书时，他大部分时间都不在家中，而是在大学学习，因此是非常幸运的。但是他的理解和关心深深吸引着我从事本书的编写工作，有这样一个儿子是令人非常愉快的。(伍德豪尔)

安德鲁·S·坦尼鲍姆

阿尔伯特·S·伍德豪尔

作者简介

安德鲁·坦尼鲍姆分别在麻省理工学院和加州大学伯克利分校获得学士和博士学位。他现任位于荷兰阿姆斯特丹市的 Vrije 大学计算机科学教授并领导着一个计算机系统研究小组。同时他还任一个研究并行、分布及图像系统的校际研究生院 - 计算机与图像高级学院的院长。

坦尼鲍姆先前的研究领域包括编译器、操作系统、网络和局域分布式系统，他现在的研究主要集中在可扩展到数百万用户的广域分布式系统。对这些课题的研究使他在学报和会议上发表了 70 余篇论文，并出版了五部专著。

坦尼鲍姆教授同时还主持开发了大量的软件。他是 Amsterdam 编译工具箱的总设计师，该工具箱被广泛地用来开发可移植的编译器，同时还用于 MINIX 的开发。他和他的博士研究生及程序员们一起设计了一个基于微内核的高性能分布式操作系统 - Amoeba。现在，以教学和研究为目的的用户可以从 Internet 上免费获得 MINIX 和 Amoeba 软件。

坦尼鲍姆的许多博士研究生在获得学位后都取得了非常丰硕的成果，这令坦尼鲍姆非常自豪，因为这是他诲人不倦的结果。

坦尼鲍姆教授同时还是 ACM 的会士、IEEE 高级会员、荷兰皇家艺术和科学院院士，他曾获得 1994 年 ACM Karl V. Karlstrom 杰出教育奖和 1997 年 ACM/SIGCSE 计算机科学教育杰出贡献奖。他被列入 Internet 上的 Who's Who in the World 名单，他在 WWW 上的主页地址为：http://www.cs.vu.nl/~ast/。

阿尔伯特·伍德豪尔分别在麻省理工学院和华盛顿大学获得学士和博士学位。他进麻省理工学院本来是想成为一名电气工程师，可是后来却成了生物学家。从 1973 年起他开始在位于麻省 Amherst 的 Hampshire 自然科学学院工作。当微型计算机慢慢多起来的时候，作为使用电子检测仪器的生物学家，他开始使用微型计算机。他给学生开设的检测仪器方面的课程逐渐演变为计算机接口和实时程序设计。

伍德豪尔博士对教学和科学技术的发展有浓厚的兴趣，在进入研究生院之前他曾在尼日利亚教过两年中学，近年来他曾几次利用自己的假期在尼加拉瓜教授计算机科学。

他对计算机作为电子系统，以及计算机与其他电子系统的相互配合很感兴趣。他最喜欢讲授的课程有计算机体系结构、汇编语言程序设计、操作系统和计算机通信。他还为开发电子器件及相关软件担当顾问。

在学术之外，伍德豪尔有不少兴趣，包括各种户外运动，业余无线电制作和读书。他还喜欢旅游和学习别国语言。他的 WWW 主页就存在一台运行 MINIX 的机器上，地址是：http://minix1.hampshire.edu/asw/。

目　录

第1章 引 言

计算机如果离开了软件将成为一堆废铜烂铁。有了软件，计算机可以对信息进行存储、处理和检索，可以显示多媒体文档、搜索 Internet 并完成其他工作。计算机软件大致分为两类：系统软件和应用软件。系统软件管理计算机本身及应用程序；应用软件执行用户最终所需要的功能。最基本的系统软件是**操作系统**(operating system)，它控制计算机的所有资源并提供开发应用程序的基础。

现代计算机系统包含一个或多个处理器、若干内存(常称为 RAM——随机存取存储器)、磁盘、打印机、网络接口及其他输入/输出设备。编写一个程序来管理所有这些器件以正确地使用它们，即使不考虑优化也是一件很困难的事情。如果每个程序员都必须处理磁盘如何工作，再加上每读一个磁盘块都有几十种因素可能导致操作出错，那么很多程序简直没法写。

许多年以前人们就认识到必须找到某种方法将硬件的复杂性同程序员分离开来。经过不断探索和改进，目前采用的方法是在裸机上加载一层软件来管理整个系统，同时给用户提供一个更容易理解和进行程序设计的接口，这被称为**虚拟机**(virtual machine)。这样一层软件就是操作系统。

这种处理方式如图 1-1 所示。底层是硬件，它本身可能包括两层或多层。最低一层是物理器件，包括集成电路芯片、连线、电源、监视器等，它们的构造和工作方式属于电气工程师的范围。

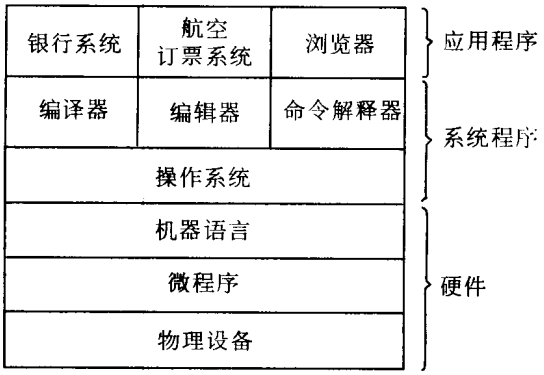

图 1-1 计算机系统由硬件、系统程序和应用程序组成

接着是**微程序**(microprogram)，通常存放在只读存储器中，它是一层很原始的软件，用来控制设备并向上一层提供一个更清晰的接口。微程序实际上是一个解释器，它先取得机器语言指令，如 ADD，MOVE 和 JUMP 等，然后通过一个动作序列来执行这些指令。例如，为了执行一条 ADD 指令，微程序必须先确定运算数据的位置，然后取数，相加，最后存放得数。由微程序解释执行的这一套指令称为机器语言。机器语言并不是硬件的组成部分，但

硬件制造商通常在手册中给出机器语言的完整描述，所以许多人将它认作真正的“计算机”。

采用**精简指令集计算机**(RISC)技术的计算机没有微程序层，其机器指令通过硬件逻辑直接执行。例如 Motorola 680x0 有微程序，而 IBM PowerPC 则没有。

机器语言典型地有 50 到 100 条指令，大多数用来完成数据传送、算术运算和数值比较等操作。在这个层次上，通过向特殊的设备寄存器写特定的数值来控制输入/输出设备。例如将磁盘地址、内存地址、读字节数和操作类型(读/写)等值写入特定的寄存器便可完成硬盘读操作。实际操作往往需要更多的参数，而操作完成后的返回状态也非常复杂。进一步而言，对于许多 I/O 设备，时序在程序设计中的作用非常重要。

操作系统的主要功能之一就是将所有这些复杂性隐藏起来，同时为程序员提供一套更加方便的指令。比如，“从文件中读一个数据块”在概念上比低层的“移动磁头臂，等待旋转延迟”之类的细节来得简单、方便。

在操作系统之上是其他系统软件，包括命令**解释器**(shell)、窗口系统、编译器、编辑器及类似的独立于应用的程序。要注意它们本身并不是操作系统的组成部分，尽管它们通常由计算机厂商提供。这一点很重要，操作系统专指在**核心态**(kernel mode)，或称**管态**(supervisor mode)下运行的软件，它受硬件保护而免遭用户的篡改。编译器和编辑器运行在**用户态**(user mode)。如果用户不喜欢某一个编译器，他可以自己重写一个，但他却不可以写一个磁盘中断处理程序，因为这是操作系统的一部分，而且硬件阻止用户对它进行修改。

系统软件之上是应用软件，这些软件可以是购买的或者是用户自行开发的，它们用来解决特定的问题，如字处理、表格处理、工程计算或者电子游戏等。

1.1 什么是操作系统

多数计算机用户都使用过操作系统，但要精确地给出操作系统的定义却很困难，部分原因是操作系统完成两项相对独立的任务，下面我们逐项进行讨论。

1.1.1 操作系统作为虚拟机

对多数计算机而言，在机器语言一级的体系结构(指令集、存储组织、I/O 和总线结构)上编程是很困难的，尤其是输入输出操作。例如考虑使用多数 PC 机采用的 NEC PD765 控制器芯片(或功能等价的芯片)来进行软盘 I/O 操作。PD765 有 16 条命令，它通过向一个设备寄存器装入特定的数据来执行这些命令。命令数据长度从 1 到 9 字节不等，其中包括：读写数据、移动磁头臂、格式化磁道、初始化、检测磁盘状态、复位、校准控制器及设备等。

最基本的命令是读数据和写数据。它们均需要 13 个参数，所有这 13 个参数被封装在 9 个字节中。这些参数指定的信息有：欲读取的磁盘块地址、每条磁道的扇区数、物理介质的数据记录格式、扇区间隙、以及对已删除数据地址标识的处理方法等。当磁盘操作结束时，控制器芯片返回 23 个状态及出错信息，它们被封装在 7 个字节中。此外，程序员还要注意步进电机的开关状态。如果电机关闭，则在读写数据前要先启动它(有一段较长的加速时间)。还要注意电机不能长时间处于开启状态，否则将损坏软盘，所以程序员必须在较长的启动延迟和可能对软盘造成损坏之间作出折衷。

显然，程序员不想涉及硬件的这些具体细节(也包括硬盘，它与软盘不同，但同样很复