国 外 计 算 机 科 学 教 材 系 列

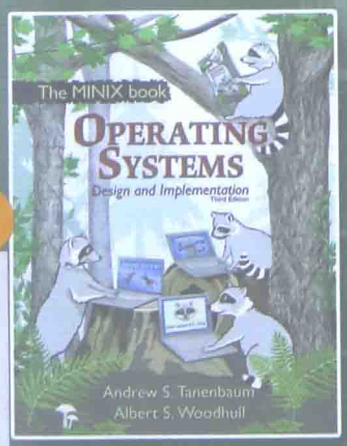
PEARSON

操作系统设计与实现

（第三版）（上册）

Operating Systems

Design and Implementation，Third Edition



[美] Andrew S. Tanenbaum
Albert S. Woodhull 著

陈 渝 谌卫军 译
向 勇 审校

中国工信出版集团

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

国外计算机科学教材系列

操作系统设计与实现

（第三版）（上册）

Operating Systems

Design and Implementation, Third Edition

[美] Andrew S. Tanenbaum 著
Albert S. Woodhull

陈 渝 谌卫军 译

向 勇 审校

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书是操作系统领域的权威教材之一。全书详细介绍了操作系统的基本原理，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、存储管理、调页算法、文件系统设计、安全和保护机制等，并深入讨论了MINIX 3操作系统。这种安排不仅可让读者了解操作系统的基本原理，而且可让读者了解如何将基本原理应用到真实的操作系统中去。

本书适用于高校计算机专业的学生，也可供程序设计人员、工程技术人员、系统架构师等相关人员参考使用。

版权贸易合同登记号　图字：01-2006-0890

图书在版编目（CIP）数据

操作系统设计与实现：第3版．上册/（美）塔嫩鲍姆（Tanenbaum, A. S.）等著；陈渝，谌卫军译．
北京：电子工业出版社，2015.6
书名原文：Operating Systems: Design and Implementation, Third Edition
国外计算机科学教材系列
ISBN 978-7-121-26193-0

I. ①操…　II. ①塔…　②陈…　③谌…　III. ①操作系统－程序设计－高等学校－教材　IV. ①TP316

中国版本图书馆CIP数据核字（2015）第114577号

策划编辑：谭海平
责任编辑：谭海平
印　　刷：北京市海淀区四季青印刷厂
装　　订：三河市皇庄路通装订厂
出版发行：电子工业出版社
　　北京市海淀区万寿路173信箱　邮编　100036
开　　本：787×1 092　1/16　印张：29.5　字数：830千字
版　　次：2015年6月第1版（原著第3版）
印　　次：2015年6月第1次印刷
定　　价：69.00元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线：（010）88258888。

出 版 说 明

21世纪初的5至10年是我国国民经济和社会发展的重要时期，也是信息产业快速发展的关键时期。在我国加入WTO后的今天，培养一支适应国际化竞争的一流IT人才队伍是我国高等教育的重要任务之一。信息科学和技术方面人才的优劣与多寡，是我国面对国际竞争时成败的关键因素。

当前，正值我国高等教育特别是信息科学领域的教育调整、变革的重大时期，为使我国教育体制与国际化接轨，有条件的高等院校正在为某些信息学科和技术课程使用国外优秀教材和优秀原版教材，以使我国在计算机教学上尽快赶上国际先进水平。

电子工业出版社秉承多年来引进国外优秀图书的经验，翻译出版了“国外计算机科学教材系列”丛书，这套教材覆盖学科范围广、领域宽、层次多，既有本科专业课程教材，也有研究生课程教材，以适应不同院系、不同专业、不同层次的师生对教材的需求，广大师生可自由选择和自由组合使用。这些教材涉及的学科方向包括网络与通信、操作系统、计算机组织与结构、算法与数据结构、数据库与信息处理、编程语言、图形图像与多媒体、软件工程等。同时，我们也适当引进了一些优秀英文原版教材，本着翻译版本和英文原版并重的原则，对重点图书既提供英文原版又提供相应的翻译版本。

在图书选题上，我们大都选择国外著名出版公司出版的高校教材，如Pearson Education培生教育出版集团、麦格劳－希尔教育出版集团、麻省理工学院出版社、剑桥大学出版社等。撰写教材的许多作者都是蜚声世界的教授、学者，如道格拉斯·科默（Douglas E. Comer）、威廉·斯托林斯（William Stallings）、哈维·戴特尔（Harvey M. Deitel）、尤利斯·布莱克（Uyless Black）等。

为确保教材的选题质量和翻译质量，我们约请了清华大学、北京大学、北京航空航天大学、复旦大学、上海交通大学、南京大学、浙江大学、哈尔滨工业大学、华中科技大学、西安交通大学、国防科学技术大学、解放军理工大学等著名高校的教授和骨干教师参与了本系列教材的选题、翻译和审校工作。他们中既有讲授同类教材的骨干教师、博士，也有积累了几十年教学经验的老教授和博士生导师。

在该系列教材的选题、翻译和编辑加工过程中，为提高教材质量，我们做了大量细致的工作，包括对所选教材进行全面论证；选择编辑时力求达到专业对口；对排版、印制质量进行严格把关。对于英文教材中出现的错误，我们通过与作者联络和网上下载勘误表等方式，逐一进行了修订。

此外，我们还将与国外著名出版公司合作，提供一些教材的教学支持资料，希望能为授课老师提供帮助。今后，我们将继续加强与各高校教师的密切联系，为广大师生引进更多的国外优秀教材和参考书，为我国计算机科学教学体系与国际教学体系的接轨做出努力。

电子工业出版社

教材出版委员会

主　任	杨芙清	北京大学教授 中国科学院院士 北京大学信息与工程学部主任 北京大学软件工程研究所所长
委　员	王　珊	中国人民大学信息学院院长、教授
	胡道元	清华大学计算机科学与技术系教授 国际信息处理联合会通信系统中国代表
	钟玉琢	清华大学计算机科学与技术系教授、博士生导师 清华大学深圳研究生院信息学部主任
	谢希仁	中国人民解放军理工大学教授 全军网络技术研究中心主任、博士生导师
	尤晋元	上海交通大学计算机科学与工程系教授 上海分布计算技术中心主任
	施伯乐	上海国际数据库研究中心主任、复旦大学教授 中国计算机学会常务理事、上海市计算机学会理事长
	邹　鹏	国防科学技术大学计算机学院教授、博士生导师 教育部计算机基础课程教学指导委员会副主任委员
	张昆藏	青岛大学信息工程学院教授

译 者 序

Andrew S. Tanenbaum是著名的计算机科学家、教育家，荷兰皇家科学艺术院院士，也是IEEE会士和ACM会士，目前供职于荷兰阿姆斯特丹Vrije大学。他在操作系统、分布式系统和计算机网络等领域都有很深的造诣，曾多次获奖，包括1994年度ACM Karl V. Karlstrom杰出教育家奖、1997年度ACM CSE计算机科学教育杰出贡献奖、2002年度TAA优秀教材奖和2003年度TAA McGuffey奖。

20世纪80年代，出于教学工作的需要，Tanenbaum教授开发了一个小巧、完整、开放源代码、UNIX兼容的操作系统MINIX，使学生可以通过剖析这个“麻雀虽小，五脏俱全”的系统，来研究其内部的运作机理。为了便于学习，他还出版了相应的教材，即本书的第一版。经过20多年的发展，MINIX系统在许多方面得到了改进，如对现代主流硬件设备的支持、对POSIX标准的支持、微内核系统结构等；与之相对应，本书也不断推陈出新，在2006年出版了第三版。

本书的最大特点就是理论与实践的完美结合。在多年的教学实践中，我们深刻地体会到，对于操作系统这样一门实用性和实践性很强的课程，如果只是单纯地介绍它的基本原理和基本概念，很难有非常理想的教学效果。一个连进程的创建函数都没有用过的人，很难想象他能对进程与线程之间的区别有真正的了解。同样，一个没有分析过内存分配源代码的人，也很难对虚拟存储管理有太多深入的理解。而本书的出现则弥补了这个缺陷，在理论与实践之间，搭起了一座桥梁。本书涵盖了操作系统课程的所有内容，包括进程管理、存储管理、文件系统和设备管理等。对于每一个章节，在组织结构上采用了从浅到深、从抽象到具体、从宏观到细节的讲授方式。首先从总体上介绍操作系统的基本原理和基本概念，然后结合MINIX 3系统，深入探讨这些基本原理的具体实现过程，最后再以源代码的形式给出了所有的实现细节。通过这种自顶向下、逐步求精的学习过程，使读者能够做到融会贯通。在面对抽象、枯燥的理论时，能够用技术实现来加以印证、加深理解；在面对复杂、繁琐的源代码时，能够用理论思想来进行指导。相信这样的一种学习模式，对于读者深入掌握操作系统的原理、设计与实现，是大有裨益的。

本书的另一个特点是实用性。如果说MINIX 1和MINIX 2还主要是用于教学目的，那么MINIX 3则完全不同。它的设计目标是一个实用的、具有高可靠性、灵活性和安全性的系统，能够运行在一些资源有限或者是嵌入式的硬件平台上。系统采用微内核结构，内核代码仅有4000行左右，而设备驱动程序等模块则作为普通的用户进程运行，这种结构大大提高了系统的可靠性，读者只要加以修改，就可以移植到自己的硬件平台上。

基于上述原因，我们认为翻译本书、把它介绍给国内的读者是一件非常有意义的事情，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2章、第3章由陈渝翻译并统稿，第1章、第4章、第5章、第6章由谌卫军翻译，向勇对全书进行了审校。在整个翻译过程中，清华大学计算机系和软件学院的师生给予了许多帮助，并且在计算机系和软件学院的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。

译者

2006年11月于清华园

本书定稿前的部分内容得到了广州海洋大学计算机系主任高升教授的审阅——编者注。

作者简介

Andrew S. Tanenbaum

Andrew S. Tanenbaum分别在麻省理工学院和加州大学伯克利分校获得学士与博士学位。现任荷兰阿姆斯特丹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。到2005年1月卸任为止，他担任计算与成像高级学院院长一职已有12年。

Tanenbaum过去的研究领域包括编译器、操作系统、网络和局域分布式系统，而现在的研究方向则主要为计算机安全，尤其是操作系统、网络以及分布式系统的安全。在所有这些研究领域，Tanenbaum发表了超过100篇论文，并出版了5本书籍。

Tanenbaum教授还编写了大量软件。他是Amsterdam Compiler Kit（一种广泛使用的、用于编写可移植编译器以及MINIX的工具集）的主要开发者，而该系统则是Linux诞生的灵感与基础。与他的博士生及程序员一起，他帮助设计了Amoeba分布式操作系统（一个基于微内核的、高性能局域分布式操作系统）。此后，他是Globe（一个可处理10亿用户的广域分布式操作系统）的设计者之一。所有这些软件现在均可在互联网上免费获得。

他的博士生在毕业后均取得了很大的成绩，他为此感到非常骄傲。

Tanenbaum教授是ACM会士、IEEE会士以及荷兰皇家科学艺术院院士。他还是1994年度ACM Karl V. Karlstrom杰出教育家奖的获得者，1997年度ACM/SIGCSE计算机科学教育杰出贡献奖的获得者，以及2002年度优秀教材奖的获得者。2004年，他被推选为荷兰皇家学会的五位新学会教授之一。他的主页地址为 *http://www.cs.vu.nl/~ast/*。

Albert S. Woodhull

Albert S. Woodhull在麻省理工学院获得学士学位，在华盛顿大学获得博士学位。在入读麻省理工学院时，他希望能成为一位电气工程师，但最后却成了一位生物学家。20多年来，他一直工作于麻省Amherst的Hampshire自然科学院。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。他给学生开设的仪器检测方面的课程逐渐演变为计算机接口和实时程序设计课程。

Woodhull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讲授计算机科学。

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任顾问。

在学术之外，Woodhull还有不少兴趣，包括各种户外活动、业余无线电制作和读书。他还喜欢旅游和学习别国语言。他的主页放在一台运行MINIX的机器上，地址为 *http://minix1.hampshire.edu/asw/*。

序　言

大多数关于操作系统的图书均重理论而轻实践，而本书则在这两者之间进行了较好的折中。本书详细探讨了操作系统的基本原理，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、存储管理、调页算法、文件系统设计、安全和保护机制等；此外，还详细讨论了一个特殊的操作系统MINIX 3（一个与UNIX兼容的操作系统），甚至提供了该系统的源代码，以便于读者仔细研究。这种安排不仅可让读者了解操作系统的基本原理，而且可让读者了解到这些基本原理是如何应用到真实的操作系统中去的。

自本书第一版于1987年推出以来，操作系统课程的教学方式已有了较小的革新。在此之前，大多数课程都只讲授理论知识。随着MINIX的出现，许多学校开始开设实验课程，以便学生仔细研究操作系统，了解其内部工作机理。对于这种教学方式，我们相当欣慰并希望能继续加强这种趋势。

MINIX在推出以来的10年里经历了许多变化。最初的代码是为基于8088的有着256 KB内存和两个磁盘驱动器的IBM PC机（没有硬盘）开发的，它基于UNIX V7。随着时间的推移，MINIX在许多方面已得到了改进，例如它支持有着大内存和多个硬盘的32位保护模式的机器，而且它不再基于UNIX V7，而基于国际POSIX标准（IEEE 1003.1和ISO 9945-1）。与此同时，UNIX中添加了许多新特性，在我们看来，这些新添加的特性可能很多，但在某些人看来这些新特性还不够，因而导致了Linux的出现。此外，MINIX还被移植到了许多其他的平台，包括Macintosh, Amiga, Atari和SPARC。本书的第二版于1997年出版并已在大学中广泛使用。

MINIX仍在得到人们的欢迎，这可以通过Google上对MINIX的搜索次数看出。

本书的第三版已有了很大的变化。我们修订了关于原理的几乎所有内容，并添加了许多新内容，但主要变化是对新操作系统MINIX 3的讨论以及本书中包含的新代码。尽管仍松散地基于MINIX 2，但MINIX 3在几个主要的方面几乎完全不同。

设计MINIX 3的目的基于如下事实：操作系统正变得越来越大、越来越慢以及越来越不可靠。与其他电子设备如电视、手机以及DVD播放器相比，操作系统变得越来越容易崩溃，加之具有许多特性和选项，实际上几乎没有人能完全理解它们，或将它们管理得很好。当然，计算机病毒、蠕虫、间谍软件以及其他形式的恶意程序也已变得越来越猖獗，这对操作系统无疑会造成较大的影响。

在很大程度上，这些问题均是由当前操作系统中的基本设计缺陷引起的：模块性的缺失。整个操作系统一般会由几百万行C/C++代码组成，这些代码被编译到了一个在内核态下运行的巨大可执行程序中。在几百万行代码中，即使只有一行代码存在缺陷，也会导致系统发生故障。使所有这些代码均正确是不可能的，尤其是当70%的代码是由设备驱动程序组成时，因为这些设备驱动程序是由第三方编写的，而这已超出了操作系统编写者的控制范围。

通过使用MINIX 3，我们证明了在设计操作系统时，并不是只有整体式设计这一种方法。MINIX 3内核仅包含有约4000行可执行代码，而不是Windows, Linux, Mac OS X或FreeBSD的数百万行代码。系统中的其他内容，包括所有的设备驱动程序（除时钟驱动程序外），是一个模块化的用户模式进程的小集合，对于每个进程，我们只关注其作用以及与之通信的其他进程。

尽管MINIX 3仍在改进，但我们相信这个将操作系统构建为高度封装的用户模式进程的模型完全可以在将来用于构建更为可靠的系统。MINIX 3适用于小型PC机（如那些可在第三世界国家

找到的机器以及嵌入式系统，这些机器的资源均有限）。无论如何，这种设计可使得学生更易于了解操作系统的工作原理，而不必去试图研究巨大的整体式操作系统。

本书附带的CD-ROM是一张可引导CD。读者可将该CD-ROM放到驱动器中，重新启动计算机，之后MINIX 3会在几秒钟后显示登录提示符。读者可登录为root用户来启动系统，而不用将MINIX 3首先安装到硬盘中。当然，MINIX 3也可安装到硬盘上。详细的安装信息请参阅附录A。

像上面建议的那样，MINIX 3正在快速发展，新版本正在频繁地推出。要下载当前的CD-ROM映像文件来刻录光盘，请进入MINIX的官方网站 *www.minix3.org*。该网站还包含有大量的新软件、文献，以及关于MINIX 3开发的新闻。关于MINIX 3的讨论以及频繁问及的问题，读者可进入USENET新闻组 *comp.os.minix* 了解详情。没有新闻阅读器的人们可按网站 *http://groups.google.com/group/comp.os.minix* 上的说明操作。

MINIX 3除了可安装到硬盘上来运行外，也可在几个PC模拟器上运行。网站的主页上列出了一些模拟器。

使用本书作为教材的教师，可通过Prentice Hall出版公司在当地的机构来获得习题解答（索取方式请参阅书后的“教学支持说明”）。此外，本书有其自己的配套网站，网址为 *www.prenhall.com/tanenbaum*。

在本书的编写过程中，我们有幸得到了许多人的帮助。Ben Gras和Jorrit Herder在时间仓促的情况下，完成了新版本操作系统的大部分编程工作；此外，他们还阅读了本书的初稿并做了许多有用的注释。在此，我们要对他们深表谢意。

Kees Bot曾为MINIX 2做了许多工作，这为我们开发MINIX 3奠定了很好的基础。Kees为MINIX的最初版本至版本2.0.4编写了大量的代码，修复了缺陷，并回答了许多问题。Philip Homburg也编写了大量的代码，并在其他方面也给予了我们大量的帮助，尤其是在初稿的审读方面。

当然，MINIX的推出还得到了其他许多人的帮助，在此我们一并表示致谢。

一些人阅读了部分初稿并提出了修订建议。在此要特别感谢Gojko Babic, Michael Crowley, Joseph M. Kizza, Sam Kohn Alexander Manov和Du Zhang。

最后，我们要感谢我们的家庭。Suzanne已是第16次在我埋头写作时给予我支持，Barbara已是第15次，而Marvin已是14次。他们的支持和爱心对我非常重要（AST）。

Al's Barbara已是第二次在我写作时给我支持。若没有她的支持、耐心，我们是不可能完成这项工作的。我的儿子Gordon一直是一位有耐心的倾听者，能得到儿子的理解与支持，对我而言无疑十分欣慰。最后，教父Zain的生日恰好就是MINIX 3的发布日期，也许某一天他会对此感到高兴的（ASW）。

Andrew S. Tanenbaum
Albert S. Woodhull

目　录

第 1 章　引言 1
1.1　什么是操作系统 3
1.1.1　操作系统作为扩展机 3
1.1.2　操作系统作为资源管理器 3
1.2　操作系统的发展历史 4
1.2.1　第一代计算机（1945-1955）：真空管和插接板 5
1.2.2　第二代计算机（1955-1965）：晶体管和批处理系统 5
1.2.3　第三代计算机（1965-1980）：集成电路和多道程序 6
1.2.4　第四代计算机（1980-）：个人计算机 10
1.2.5　MINIX 3 的历史 11
1.3　操作系统概念 13
1.3.1　进程 14
1.3.2　文件 15
1.3.3　命令解释器 17
1.4　系统调用 18
1.4.1　进程管理的系统调用 20
1.4.2　信号管理的系统调用 22
1.4.3　文件管理的系统调用 24
1.4.4　目录管理的系统调用 27
1.4.5　保护的系统调用 29
1.4.6　时间管理的系统调用 30
1.5　操作系统结构 30
1.5.1　整体结构 31
1.5.2　分层结构 33
1.5.3　虚拟机 33
1.5.4　外核 35
1.5.5　客户 - 服务器模型 36
1.6　剩余各章内容简介 37
1.7　小结 37
习题 37

第 2 章　进程 39
2.1　进程介绍 39
2.1.1　进程模型 39
2.1.2　进程的创建 40

2.1.3 进程的终止 41
2.1.4 进程的层次结构 42
2.1.5 进程的状态 43
2.1.6 进程的实现 44
2.1.7 线程 45
2.2 进程间通信 48
2.2.1 竞争条件 48
2.2.2 临界区 49
2.2.3 忙等待形式的互斥 50
2.2.4 睡眠和唤醒 53
2.2.5 信号量 55
2.2.6 互斥 57
2.2.7 管程 57
2.2.8 消息传递 60
2.3 经典 IPC 问题 62
2.3.1 哲学家进餐问题 62
2.3.2 读者 - 写者问题 65
2.4 进程调度 66
2.4.1 调度介绍 66
2.4.2 批处理系统中的调度 69
2.4.3 交互式系统中的调度 72
2.4.4 实时系统调度 76
2.4.5 策略与机制 76
2.4.6 线程调度 77
2.5 MINIX 3 进程概述 78
2.5.1 MINIX 3 的内部结构 78
2.5.2 MINIX 3 中的进程管理 80
2.5.3 MINIX 3 中的进程间通信 83
2.5.4 MINIX 3 中的进程调度 85
2.6 MINIX 3 中进程的实现 86
2.6.1 MINIX 3 源代码的组织 86
2.6.2 编译及运行 MINIX 3 88
2.6.3 公共头文件 90
2.6.4 MINIX 3 头文件 95
2.6.5 进程数据结构和头文件 101
2.6.6 引导 MINIX 3 107
2.6.7 系统初始化 110
2.6.8 MINIX 的中断处理 114
2.6.9 MINIX 3 的进程间通信 121
2.6.10 MINIX 的进程调度 124

2.6.11 与硬件相关的内核支持 126
2.6.12 实用程序和内核库 129
2.7 MINIX 3 的系统任务 131
2.7.1 系统任务综述 132
2.7.2 系统任务的实现 134
2.7.3 系统库的实现 136
2.8 MINIX 3 的时钟任务 138
2.8.1 时钟硬件 139
2.8.2 计时程序 140
2.8.3 MINIX 3 中的时钟驱动程序总览 142
2.8.4 MINIX 3 中的时钟驱动程序的应用 144
2.9 小结 145
习题 146

第 3 章 输入 / 输出系统 150
3.1 I/O 硬件原理 150
3.1.1 I/O 设备 150
3.1.2 设备控制器 151
3.1.3 内存映射 I/O 152
3.1.4 中断 153
3.1.5 直接存储器存取 154
3.2 I/O 软件的原理 155
3.2.1 I/O 软件的目标 155
3.2.2 中断处理器 156
3.2.3 设备驱动程序 157
3.2.4 与设备无关的 I/O 软件 158
3.2.5 用户空间的 I/O 软件 160
3.3 死锁 161
3.3.1 资源 161
3.3.2 死锁的原理 162
3.3.3 鸵鸟算法 165
3.3.4 死锁的检测和恢复 166
3.3.5 死锁的预防 166
3.3.6 避免死锁 168
3.4 MINIX 3 中的 I/O 概述 171
3.4.1 MINIX 3 中的中断处理器和 I/O 访问 171
3.4.2 MINIX 3 的设备驱动程序 173
3.4.3 MINIX 3 中与设备无关的 I/O 软件 176
3.4.4 MINIX 3 中的用户级 I/O 软件 176
3.4.5 MINIX 3 的死锁处理 177
3.5 MINIX 3 中的块设备 177

3.5.1 MINIX 3 中的块设备驱动程序概述 177
3.5.2 通用块设备驱动程序软件 180
3.5.3 驱动程序库 182
3.6 RAM 盘 183
3.6.1 RAM 盘硬件和软件 184
3.6.2 MINIX 3 中的 RAM 盘驱动程序概述 185
3.6.3 MINIX 3 中 RAM 盘驱动程序的实现 186
3.7 磁盘 188
3.7.1 磁盘硬件 188
3.7.2 RAID 189
3.7.3 磁盘软件 190
3.7.4 MINIX 3 中的硬盘驱动程序简介 194
3.7.5 MINIX 3 中硬盘驱动程序的实现 196
3.7.6 软盘处理 203
3.8 终端 204
3.8.1 终端硬件 205
3.8.2 终端软件 208
3.8.3 MINIX 3 中的终端驱动程序简介 213
3.8.4 设备无关终端驱动程序的实现 224
3.8.5 键盘驱动程序的实现 236
3.8.6 显示驱动程序的实现 241
3.9 小结 246
习题 247

第 4 章 存储管理 251
4.1 基本的存储管理 251
4.1.1 单道程序存储管理 252
4.1.2 固定分区的多道程序系统 252
4.1.3 重定位和存储保护 254
4.2 交换技术 255
4.2.1 基于位图的存储管理 257
4.2.2 基于链表的存储管理 257
4.3 虚拟存储管理 259
4.3.1 虚拟页式存储管理 260
4.3.2 页表 263
4.3.3 关联存储器 TLB 266
4.3.4 反置页表 268
4.4 页面置换算法 269
4.4.1 最优页面置换算法 270
4.4.2 最近未使用页面置换算法 270
4.4.3 先进先出页面置换算法 271

4.4.4 第二次机会页面置换算法 271
4.4.5 时钟页面置换算法 272
4.4.6 最近最久未使用页面置换算法 273
4.4.7 LRU 算法的软件模拟 273
4.5 页式存储管理中的设计问题 275
4.5.1 工作集模型 275
4.5.2 局部与全局分配策略 277
4.5.3 页面大小 279
4.5.4 虚拟存储器接口 280
4.6 段式存储管理 281
4.6.1 纯分段系统的实现 283
4.6.2 段页式存储管理：Intel Pentium 284
4.7 MINIX 3 进程管理器概述 287
4.7.1 内存布局 288
4.7.2 消息处理 291
4.7.3 进程管理的数据结构和算法 292
4.7.4 FORK, EXIT 和 WAIT 系统调用 296
4.7.5 EXEC 系统调用 297
4.7.6 BRK 系统调用 300
4.7.7 信号处理 300
4.7.8 其他的系统调用 306
4.8 MINIX 3 进程管理器的实现 306
4.8.1 头文件和数据结构 306
4.8.2 主程序 309
4.8.3 FORK, EXIT 和 WAIT 的实现 312
4.8.4 EXEC 的实现 314
4.8.5 BRK 的实现 317
4.8.6 信号处理的实现 317
4.8.7 其他系统调用的实现 323
4.8.8 内存管理工具 326
4.9 小结 327
习题 327

第 5 章 文件系统 331
5.1 文件 331
5.1.1 文件的命名 332
5.1.2 文件的结构 333
5.1.3 文件的类型 334
5.1.4 文件的访问 336
5.1.5 文件的属性 336
5.1.6 文件的操作 337

5.2 目录 338
5.2.1 简单的目录系统 338
5.2.2 层状目录系统 339
5.2.3 路径名 340
5.2.4 目录的操作 342
5.3 文件系统的实现 342
5.3.1 文件系统的布局 342
5.3.2 文件的实现 344
5.3.3 目录的实现 347
5.3.4 磁盘空间管理 351
5.3.5 文件系统的可靠性 354
5.3.6 文件系统的性能 359
5.3.7 日志结构的文件系统 362
5.4 文件系统的安全性 363
5.4.1 安全环境 364
5.4.2 通常的安全攻击 367
5.4.3 安全性的设计原则 368
5.4.4 用户认证 368
5.5 保护机制 371
5.5.1 保护域 371
5.5.2 访问控制列表 373
5.5.3 权能 375
5.5.4 秘密通道 377
5.6 MINIX 3 文件系统概述 379
5.6.1 消息 380
5.6.2 文件系统的布局 381
5.6.3 位图 383
5.6.4 i 节点 384
5.6.5 块高速缓存 386
5.6.6 目录和路径 387
5.6.7 文件描述符 389
5.6.8 文件锁 390
5.6.9 管道和设备文件 391
5.6.10 一个例子：READ 系统调用 392
5.7 MINIX 3 文件系统的实现 392
5.7.1 头文件和全局数据结构 393
5.7.2 表格管理 395
5.7.3 主程序 401
5.7.4 对单个文件的操作 404
5.7.5 目录和路径 410
5.7.6 其他的系统调用 412

5.7.7 I/O 设备接口 414
5.7.8 附加的系统调用支持 418
5.7.9 文件系统的实用程序 419
5.7.10 其他的 MINIX 3 组件 420
5.8 小结 420
习题 421
第 6 章 阅读材料和参考文献 424
6.1 推荐的进一步阅读材料 424
6.1.1 介绍和概论 424
6.1.2 进程 426
6.1.3 输入 / 输出 426
6.1.4 存储管理 427
6.1.5 文件系统 427
6.2 按字母顺序排列的参考文献 428
索引 436

第1章 引 言

1.1 什么是操作系统
1.2 操作系统的发展历史
1.3 操作系统概念
1.4 系统调用
1.5 操作系统结构
1.6 剩余各章内容简介
1.7 小结

众所周知，软件是计算机系统的灵魂，没有软件的计算机就像一个漂亮的花瓶，虚有其表。有了软件，计算机可以存储、处理和检索信息，可以播放音乐和电影，可以发送电子邮件、搜索Internet，可以做许许多多有意思的事情。计算机软件大致可以分为两类，即系统软件和应用软件。系统软件负责管理计算机本身的运作，而应用软件则负责完成用户所需要的各种功能。最基本的系统软件是**操作系统**（Operating System, OS），它负责管理计算机的所有资源并提供一个可以在其上编写应用程序的平台。本书讨论的就是操作系统这种软件，我们将以MINIX 3系统为模型，来阐述操作系统的设计原理，以及如何来实现一个真正的系统。

现代计算机系统中包含有各种不同类型的硬件设备，如处理器（一个或多个）、内存、磁盘、打印机、键盘、显示器、网络接口以及其他的输入/输出设备。总之，这是一个复杂的系统。在这种情形下，如果要编写程序来管理所有的这些组件并正确地使用它们（暂且不论性能的优化），则将是一件非常困难的事情。因此，对于每一个程序员来说，当他在编写程序的时候，如果要考虑到该方面的所有问题，例如磁盘驱动器的工作原理、在读取一个磁盘数据块时哪些环节容易发生错误等，那么许多程序基本上无法编写。

在许多年以前，人们就已经认识到必须找到某种方法，让程序员从复杂、繁琐的硬件操作中解脱出来。经过不断探索和改进，目前的做法是在裸机上引入一层软件，让它来管理系统的各个部件，并给上层的用户提供一个易于理解和编程的接口［或者称为**虚拟机**（virtual machine）］。这样的一层软件就是操作系统。

操作系统的定位如图1.1所示。在计算机系统的底层是硬件。在许多情形下，硬件本身又可以分为两层或多层。最底层是物理设备，包括集成电路芯片、线路、电源、阴极射线管等。至于物理设备的内部结构和工作原理，则属于电气工程的范畴，本书不予详述。

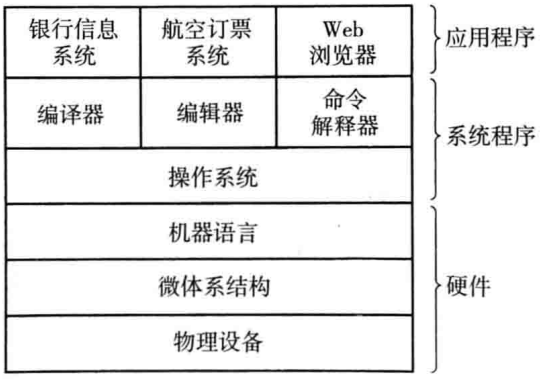


图1.1 计算机系统由硬件、系统程序和应用程序组成

接下来是**微体系结构层**（microarchitecture level）。在这一层中，各个物理设备被组织成一些功能单元，如中央处理器（Central Processing Unit, CPU）内部的一些寄存器、涉及算术逻辑单元的