

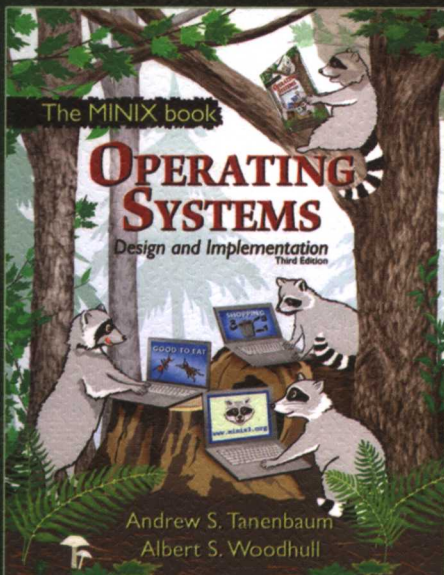
Tanenbaum's

操作系统设计与实现

(第三版) 下册

Operating Systems

Design and Implementation, Third Edition



[美] Andrew S. Tanenbaum 著
Albert S. Woodhull
陈 渝 湛卫军 译
向 勇 审校

内 容 简 介

本书是关于操作系统的权威教材,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。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。本册给出了上册中提到的MINIX 3操作系统的安装方法以及详细的原代码,并附有光盘。

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。

Simplified Chinese edition Copyright © 2006 by PEARSON EDUCATION ASIA LIMITED and Publishing House of Electronics Industry.

Operating Systems: Design and Implementation, Third Edition, ISBN: 0131429388 by Andrew S. Tanenbaum and Albert S. Woodhull. Copyright © 2006.

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Prentice Hall.

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macau).

本书中文简体字翻译版由电子工业出版社和Pearson Education培生教育出版亚洲有限公司合作出版。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有Pearson Education培生教育出版集团激光防伪标签,无标签者不得销售。

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.下册:第3版/(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,2007.3

(国外计算机科学教材系列)

书名原文:Operating Systems: Design and Implementation, Third Edition

ISBN 978-7-121-02648-5

I. 操... II. ①塔... ②陈... ③谌... III. 操作系统—教材 IV. TP316

中国版本图书馆CIP数据核字(2006)第130357号

责任编辑:谭海平

印刷:北京市海淀区四季青印刷厂

装订:北京牛山世兴印刷厂

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:787×1092 1/16 印张:23.25 字数:654千字

印次:2007年3月第1次印刷

定价:49.00元(附光盘1张)

凡所购买电子工业出版社的图书有缺损问题,请向购买书店调换;若书店售缺,请与本社发行部联系。联系电话:(010)68279077 邮购电话:(010)88254888。

质量投诉请发邮件至zlt@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关键时期,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的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

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

关于 MINIX 3 CD

系统需求

下面是安装光盘中提供的软件时的最小系统需求。

硬件

MINIX 3 操作系统要求如下的硬件：

- Pentium 或兼容处理器的 PC
- 16 MB 或更大的 RAM
- 200 MB 的可用磁盘空间
- IDE CD-ROM 驱动器
- IDE 硬盘

不支持串行 ATA, USB 和 SCSI 硬盘。对于可选择的其他配置，请访问 <http://www.minix3.org>。

软件

MINIX 3 OS 是一个操作系统。如果读者希望保留原有的操作系统和数据（建议）并创建一个双重引导机器，那么需要对硬盘分区。这时可使用如下软件之一来进行分区：

- Partition Magic（<http://www.powerquest.com/partitionmagic>）
- Partion Resizer（<http://www.zeleps.com>）
- 按网页 <http://www.minix3.org/partitions.html> 上的步骤操作

安装

在未连接到 Internet 的情况下可以完成安装，但一些高级文档仅联机出现在网站 <http://www.minix3.org> 上。完成安装的说明在 CD 上以 Adobe Acrobat PDF 格式提供。

产品支持

关于该光盘中 MINIX 软件的进一步技术信息，请访问 MINIX 的官方网站 <http://www.minix3.org>。

目 录

附录 A 安装 MINIX 3	1
附录 B MINIX 源码	7
附录 C 文件索引	356

附录 A 安装 MINIX 3

- A.1 准备
- A.2 启动
- A.3 安装到硬盘
- A.4 测试
- A.5 使用模拟器

本附录解释如何安装 MINIX 3。完全安装 MINIX 3 的要求如下：奔腾（或兼容的）处理器，至少 16 MB 的 RAM，1 GB 的硬盘可用空间，一台 IDE CD-ROM 以及一个 IDE 硬盘。最小安装（无命令源代码）需要 8 MB 的 RAM 和 50 MB 的硬盘空间。当前不支持串行 ATA、USB 和 SCSI 磁盘。对于 USB CD-ROM，请参阅网站 www.minix3.org。

A.1 准备

如果已有 CD-ROM（如取自本书），那么可跳过步骤 1 和步骤 2，但建议读者最好查阅网站 www.minix3.org，以了解是否有更新的版本可用。如果在模拟器而不是本地机器上运行 MINIX 3，则要首先参阅第五部分。如果没有 IDE CD-ROM，则要么获得一个专用的 USB CD-ROM 启动映像，要么使用一个模拟器。

1. 下载 MINIX 3 CD-ROM 映像

读者可从 MINIX 3 的网站 www.minix3.org 上下载 MINIX 3 CD-ROM 映像。

2. 下载可启动的 MINIX 3 CD-ROM

解压缩下载后的文件。这时会得到带有扩展名 *.iso* 的一个 CD-ROM 映像文件以及本手册。扩展名为 *.iso* 的文件是一个逐位 CD-ROM 映像，将其刻录在 CD-ROM 上是为了使该 CD-ROM 成为可启动的 CD-ROM。

如果正在使用 *Easy CD Creator 5* 刻录程序，那么可以从 File 菜单中选择 Record CD from CD image，并在出现的对话框中将文件类型从 *.cif* 更改为 *.iso*。选择该映像文件并单击 Open，然后单击 Start Recording。

如果正使用 *Nero Express 5* 刻录程序，那么可选择 Disc Image or Saved Project 并将类型更改为 Image Files，选择该映像文件并单击 Open。选择自己的 CD 刻录机并单击 Next。

如果正在运行 Windows XP 且没有 CD-ROM 刻录程序，那么可进入网页 alexfeinman.brinkster.net/isorecorder.htm 下载一个免费的程序并使用它来创建一个 CD 映像。

3. 确定所拥有的以太网芯片

MINIX 3 支持几种以太网芯片，包括 Intel Pro/100、RealTek 8029 和 8139、AMD LANCE 以及几种 3COM 芯片。在安装过程中，系统将会询问读者的以太网芯片是哪一种。请读者查询文档以确定这一点。另外，如果读者正在使用 Windows，那么可按如下方式进入设备管理器：

Windows 2000: Start > Settings > Control Panel > System > Hardware > Device Manager

Windows XP: Start > Control Panel > System > Hardware > Device Manager

其中 System 要求双击；其他图标或菜单项要求单击。单击 + 以展开到出现 Network adapters，这时会看到所拥有的以太网芯片类型。如果没有支持的芯片，那么仍可运行 MINIX 3，只是这时系统不支持以太网。

4. 对硬盘进行分区

如果愿意，可从CD-ROM启动计算机，MINIX 3将会启动，但如果要做一些有意义的事情，则必须在硬盘上为其创建一个分区。但在进行分区之前，为安全起见，一定要将自己的数据备份到外部媒介如CD-ROM或DVD中。读者的文件是有价值的，一定要保护好它们。

如果不知道如何对磁盘分区或是经验不足，那么这里建议读者在www.minix3.org/doc/partitions.html处阅读关于磁盘分区的联机教程。如果已经知道如何管理分区，那么创建一个至少50 MB的连续可用磁盘空间，在需要命令源代码的情况下，则至少要创建一个1 GB大小的磁盘空间。如果不知道如何管理分区但又有一个如Partition Magic这样的分区程序，那么可以使用它来创建一个有着可用磁盘空间的分区。注意，一定要确保至少有一个主分区（如Master Boot Record插槽）是可用的。MINIX 3安装脚本将指导读者在可用空间中创建一个MINIX分区，这个分区可在第一个或第二个IDE磁盘上。

如果正在运行Windows 95, 98, Me或2000，且磁盘由单个FAT分区组成，则可使用CD-ROM上的presz134.exe程序（也可在网站zeleps.com处获得）来调整其大小，以便为MINIX腾出空间。在其他情形下，请阅读上面给出的联机教程。

如果读者的磁盘大于128 GB，则MINIX 3分区必须完全位于前128 GB中（因为磁盘块是按这种方法寻址的）。

警告：如果在磁盘分区过程中出现了错误，那么可能会丢失磁盘上的所有数据，因此一定要在分区前将磁盘上的数据备份到CD-ROM或DVD中。在进行磁盘分区时一定要小心。

A.2 启动

现在，读者应已在磁盘上分配了一些可用空间。如果现在还未做到这一点，那么请立即这样做，除非读者愿意将一个已存在的分区转换为MINIX 3分区。

1. 从CD-ROM启动

将CD-ROM插入光盘驱动器中并从它启动计算机。如果有16 MB或以上的内存，可以选择Regular；如果仅有8 MB的内存，则选择Small。如果计算机是从硬盘而不是从CD-ROM启动，那么请再次启动并进入BIOS设置程序来更改启动设备的顺序，即将CD-ROM放在硬盘的前面。

2. 作为root用户登录

当登录提示符出现时，请作为root用户登录。在成功地作为root用户登录后，会看到命令提示符（#）。这时表示读者正在完整地运行MINIX 3。如果键入

```
ls/usr/bin | more
```

则会看到可用的软件。按空格键并滚动列表。为了解程序foo的功能，可键入

```
man foo
```

也可自www.minix3.org/manpages处查阅相关的手册页面。

3. 启动安装脚本

要在硬盘上开始MINIX 3的安装，可键入

```
setup
```

在该命令及所有其他命令之后，一定要按回车键。当安装脚本在屏幕上以一个分号结束时，按回车键继续。如果屏幕突然变得空白，请按 CTRL-F3 组合键来滚动选择软件（仅在非常旧的计算机上需要这样做）。注意，CTRL-F3 意味着按下 CTRL 键的同时按下 F3 键。

A.3 安装到硬盘

以下这些步骤相应于读者屏幕上的步骤。

1. 选择键盘类型

在系统要求读者选择自己国家的键盘时，请这样做。该步骤和其他步骤在方括号中提供一个默认的选择。如果用户同意默认的选择，可按回车键。在多数步骤中，默认的选择对初学者来说都是较好的选择。

2. 选择以太网芯片

现在系统将询问用户想要安装哪种可用的以太网驱动程序。请选择一种。

3. 基本的最小或完全安装

如果磁盘空间很紧张，那么可选择 M 来进行最小安装，最小安装会安装所有的二进制码以及系统的源码，但不会安装命令的源码。对于基本的系统，50 MB 可用空间已足够了。如果有 1 GB 或大于 1 GB 的可用空间，那么可以选择 F 来进行完全安装。

4. 为 MINIX 3 创建或选择一个分区

系统首先会询问读者是否精通于为 MINIX 3 进行磁盘分区。如果精通，则可进入分区程序并为用户提供编辑主引导记录的全部权限。如果不精通，则可按回车键接受默认的操作，默认操作会为 MINIX 3 自动地逐步格式化磁盘分区。

4.1 选择一个磁盘来安装 MINIX 3

一个 IDE 控制器最多可接四个磁盘。安装脚本现在会搜索每个磁盘。这时可忽略任何错误消息。在列出驱动器后，选择其中的一个驱动器并确认自己的选择。如果有两个硬盘且决定将 MINIX 3 安装到第二个硬盘上并在第二硬盘启动时出现了问题，可参阅 www.minix3.org/doc/using2disks.html。

4.2 选择一个磁盘区域

现在选择一个区域来安装 MINIX 3。这时有三种选择：

- (1) 选择一个可用的区域。
- (2) 选择一个将要覆盖的区域。
- (3) 删除一个分区以腾出空间并与相邻的可用空间合并。

对于选择(1)和选择(2)，键入磁盘区域编号。对于选择(3)，键入

```
delete
```

然后在系统询问时给出区域编号。这个区域将被覆盖且其先前的内容将永远消失。

4.3 确认选择

现在将进入不可逆转的阶段。这时系统会询问是否继续。如果继续，则选择的区域中的数据将会永远丢失。如果确定，可键入

```
yes
```

然后按回车键。要在不更改分区表的情况下退出安装脚本，可按 CTRL-C 组合键。

5. 重新安装选项

如果选择一个已有的 MINIX 3 分区，这时会出现两个选项，即 Full install 和 Reinstall。选择前一个选项时，会删除分区中的所有内容；选择后一个选项时，则不影响已存在的 */home* 分区。这种设计意味着用户可将自己的个人文件放到 */home* 中并重新安装 MINIX 3 的一个新版本，而不会丢失读者的个人文件。

6. 选择 */home* 的大小

选取的分区将分为三个子分区，即 *root*、*/usr* 和 */home*。后者用于存放用户自己的个人文件。请指定为自己的文件保留多少分区的空间。系统会要求用户对选择进行确认。

7. 选择块大小

系统支持 1 KB、2 KB、4 KB 和 8 KB 的块大小。如果要使用一个大于 4 KB 的块大小，则必须修改一个常量并重新编译系统。如果用户的内存是 16 MB 或更大，可使用默认值（4 KB）；否则，请使用 1KB。

8. 等待坏块检测

安装脚本现在将扫描每个分区中的坏磁盘块。这需要几分钟时间，对于较大的分区可能要 10 分钟或更长的时间，请耐心等待。如果读者确认无坏块，则可按 CTRL-C 组合键中止对每个分区的扫描。

9. 等待复制文件

在扫描完成后，文件将会自动地从 CD-ROM 中复制到硬盘中。每个文件都将声明被复制。在复制完成后，MINIX 3 的安装也就完成了。键入

```
shutdown
```

关闭系统。要总是按照这种方法来停止 MINIX 3，以避免数据丢失，因为 MINIX 3 会在 RAM 磁盘上保留一些文件，尽管这些文件已复制到了硬盘中。

A.4 测试

本节描述如何测试安装、在修改安装后如何重建系统以及如何在以后启动它。为开始测试，首先要启动新的 MINIX 3 系统。例如，如果使用了控制器 0、磁盘 0 和分区 3，则可键入

```
boot c0d0p3
```

并作为 root 用户登录。在很少见的条件下，BIOS 可见的驱动器号（由引导监控程序使用）可能与 MINIX 3 使用的驱动器号不一致。首先请试用安装脚本声明的那个驱动器号。这时可创建一个 root 密码。要获得帮助，请参阅 *man passwd*。

1. 编译测试套件

要测试 MINIX 3，可在命令提示符（#）下键入

```
cd /usr/src/test  
make
```

并等待，直到它完成了所有的 40 个编译。按 CTRL-D 组合键退出登录。

2. 运行测试套件

为测试系统，登录为 bin（要求）并键入

```
cd /usr/src/test
./run
```

来运行测试程序。它们应能正确地运行，但可能需要 20 分钟至 1 小时的时间，具体要视机器的快慢而定。注意，在作为 root 运行时，需要编译测试套件，但要作为 bin 执行，以了解 stupid 位是否正确地工作。

3. 重建整个操作系统

如果所有的测试工作正确，那么现在就可以重建系统。这样做是不必要的，因为它已预建完毕；但如果读者计划修改系统，则需要知道如何重建系统。此外，重建系统可较好地测试系统是否工作正常。键入

```
cd /usr/src/tools
make
```

查看各种可用的选项。现在，通过键入

```
su
make clean
time make image
```

来建立一个新的可引导映像。这只是重建操作系统，包括所有的内核态和用户态部分。所需要的时间不会很长。如果有一个软盘驱动器，则可将一张格式化后的软盘插入到其中并键入

```
make fdboot
```

来制作一张可引导软盘，以备将来使用。

当系统询问完整的路径时，可键入

```
fd0
```

这种方法当前不适用于 USB 软盘，因为现在还没有 MINIX 3 USB 软盘驱动程序。要更新当前已安装到硬盘上的引导映像，可键入

```
make hdboot
```

4. 关闭与重启新系统

要启动新系统，首先要键入

```
shutdown
```

来关闭系统。该命令会保存一些文件并回到 MINIX 3 引导监控程序中。要大致了解引导监控程序的功能，可键入

```
help
```

对于更加详细的信息，请参阅 www.minix3.org/manpages/man8/boot.8.html。现在可取出 CD-ROM 或软盘并关闭计算机。

5. 后续启动

如果有一个软盘驱动器，则启动 MINIX 3 的最简方法是插入新的引导软盘并打开计算机的电源。这只需要几秒钟时间。也可从 MINIX 3 CD-ROM 引导，登录为 bin 并键入

```
shutdown
```

以返回到 MINIX 3 引导监控程序。现在键入

```
boot c0d0p0
```

以从控制器 0、驱动器 0、分区 0 上的操作系统映像文件启动。当然，如果将 MINIX 3 放到驱动器 0 分区 1 上，那么可使用

```
boot c0d0p1
```

依次类推。

第三种启动方法是使 MINIX 3 分区成为活动分区，并使用 MINIX 3 引导监控程序来启动 MINIX 3 或任何其他操作系统。详细内容请参阅 www.minix3.org/manpages/man8/boot.8.html。

最后，第四种方法是安装一个多启动装载程序，如 LILO 或 GRUB (www.gnu.org/software/grub)。然后，就可以很容易地启动任何操作系统。多启动装载程序的讨论超出了本指南的范围，读者可在 www.minix3.org/doc 中找到关于该主题的一些信息。

A.5 使用模拟器

运行 MINIX 3 的一种完全不同的方法是在另一个操作系统之上运行它。各种虚拟机、模拟器和仿真器可用于此目的。下面是一些最为流行的模拟器：

- VMware (www.vmware.com)
- Bochs (www.bochs.org)
- QEMU (www.qemu.org)

对于上面的每个模拟器，请读者参阅相关的文档。在模拟器上运行一个程序类似于在一台实际的机器上运行它，因此读者应返回到第一部分，获取最新的 CD-ROM 并在此处继续。

附录 B MINIX 源码

```
+++++
include/ansi.h
+++++

00000 /* The <ansi.h> header attempts to decide whether the compiler has enough
00001  * conformance to Standard C for Minix to take advantage of. If so, the
00002  * symbol _ANSI is defined (as 31459). Otherwise _ANSI is not defined
00003  * here, but it may be defined by applications that want to bend the rules.
00004  * The magic number in the definition is to inhibit unnecessary bending
00005  * of the rules. (For consistency with the new '#ifdef _ANSI' tests in
00006  * the headers, _ANSI should really be defined as nothing, but that would
00007  * break many library routines that use "#if _ANSI".)
00008
00009  * If _ANSI ends up being defined, a macro
00010  *
00011  *     _PROTOTYPE(function, params)
00012  *
00013  * is defined. This macro expands in different ways, generating either
00014  * ANSI Standard C prototypes or old-style K&R (Kernighan & Ritchie)
00015  * prototypes, as needed. Finally, some programs use _CONST, _VOIDSTAR etc
00016  * in such a way that they are portable over both ANSI and K&R compilers.
00017  * The appropriate macros are defined here.
00018  */
00019
00020 #ifndef _ANSI_H
00021 #define _ANSI_H
00022
00023 #if __STDC__ == 1
00024 #define _ANSI          31459 /* compiler claims full ANSI conformance */
00025 #endif
00026
00027 #ifdef __GNUC__
00028 #define _ANSI          31459 /* gcc conforms enough even in non-ANSI mode */
00029 #endif
00030
00031 #ifdef _ANSI
00032
00033 /* Keep everything for ANSI prototypes. */
00034 #define _PROTOTYPE(function, params)    function params
00035 #define _ARGS(params)                  params
00036
00037 #define _VOIDSTAR        void *
00038 #define _VOID            void
00039 #define _CONST           const
00040 #define _VOLATILE       volatile
00041 #define _SIZET          size_t
00042
00043 #else
00044
00045 /* Throw away the parameters for K&R prototypes. */
00046 #define _PROTOTYPE(function, params)    function()
00047 #define _ARGS(params)                  ()
00048
00049 #define _VOIDSTAR        void *
00050 #define _VOID            void
00051 #define _CONST           const
00052 #define _VOLATILE       volatile
00053 #define _SIZET          int
00054
```