DEITEL DEITEL CHOFFNES

PEARSON Prentice Hall

Operating Systems Third Edition

操作系统

(第三版)

Harvey M. Deitel
(美) Paul J. Deitel 著
David R. Choffnes
施平安 江永忠 瞿中 译

PEARSON Education

清华大学出版社

操 作 系 统

(第三版)

Harvey M. Deitel
(美) Paul J. Deitel 著
David R. Choffnes
施平安 江永忠 瞿中 译

清华大学出版社
北 京

图书在版编目(CIP)数据

操作系统(第三版)/(美)戴特尔(Deitel, H. M.)等著；施平安，江永忠，瞿中 译. —北京：清华大学出版社，2007.8

书名原文：Operating Systems, Third Edition

ISBN 978-7-302-15185-2

I. 操… II. ①戴… ②施… ③江… ④瞿… III. 操作系统 IV. TP316

中国版本图书馆 CIP 数据核字(2007)第 071281 号

责任编辑：王 军 梁卫红
装帧设计：孔祥丰
责任校对：成凤进
责任印制：何 芊
出版发行：清华大学出版社 **地 址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn **邮 编**：100084
c-service@tup.tsinghua.edu.cn
社 总 机：010-62770175 **邮购热线**：010-62786544
投稿咨询：010-62772015 **客户服务**：010-62776969
印 刷 者：北京密云胶印厂
装 订 者：三河市李旗庄少明装订厂
经 销：全国新华书店
开 本：185×260 **印 张**：86 **字 数**：2093 千字
版 次：2007 年 8 月第 3 版 **印 次**：2007 年 8 月第1 次印刷
印 数：1～3000
定 价：150.00 元

前　言

欢迎来到操作系统的世界。本书主要设计为大学一学期或两学期的操作系统课程教材，适合计算机专业低、高年级本科生或研究生使用(根据最近的 ACM/IEEE 课程的定义)。操作系统设计者和系统程序员也可将本书作为参考书使用。

本文的特点在于，它针对当今最重要的两种操作系统给出了详尽的案例分析，即 Linux 以及 Microsoft Windows XP。这两种操作系统分别代表两种不同的操作系统设计模式——自由的、开源的开发，以及需要许可证的企业开发。Linux 案例分析以 kernel release 2.6 为蓝本。Windows XP 案例分析则强调了当今最流行的个人计算机操作系统的最新版本的内部机制。通过这些案例分析，你可以加深理解现实世界的系统所采用的设计及实现原则。

无论 Linux 还是 Windows XP，它们都是大规模的、复杂的操作系统，由数百万行源代码构成。我们研究了每一种操作系统的主要组件。案例分析揭示了个人计算机、工作站、多处理器、分布式和嵌入式环境的问题。详细讨论了作为开放源码和开放系统，Linux 和其他 UNIX 操作系统为什么受到了一些大公司的青睐。

在本前言中，我们介绍了本书采用的教学方法以及本书的关键内容和设计元素，还讨论了本书的辅助支持。在“本书导读”一节，我们概括了本书要讲述的操作系统的每一个方面。

在阅读本书的过程中，如果你有任何问题，请致函 deitel@deitel.com，我们将快速做出回答。请访问我们的网站 www.deitel.com，并订阅 Deitel® Buzz Online 新闻邮件(www.deitel.com/newsletter/subscribe.html)。我们将通过网站和新闻邮件向读者提供有关所有 Deitel 出版物和服务的最新信息。除此之外，我们还为本书专门开设了一个网址，即 www.deitel.com/books/os3e。在这里，我们将面向学生和教师发布各种辅助材料，其中包括相关的图、内容更新、勘误、推荐的学生研究项目、示范教学大纲以及本书上一版提供的研究材料参考书目。Prentice Hall 也为本书提供了一个配套网站，地址是 www.prenhall.com/deitel。

本书经过严格的审稿，审稿人都是来自专业院校和业界的专业人士。在后面“致谢”一节中列出了他们的名字及所属单位。

本书特点

本书包含大量新内容。除此之外，我们还修订和更新了第二版的大量内容。本书强调了分布式计算的最新技术和问题，这使本书在同类书籍中更为出众。我们新增了大量篇幅来介绍嵌入式、实时和分布式系统。本版的一些关键特点包括：

- 符合 ACM/IEEE CC2001 操作系统课程的所有核心要求。
- 讨论了所有 CC2001 选修操作系统主题(shell scripting 除外)。
- 更新了硬件介绍，加入了对最新技术的讨论，并强调了它们对操作系统设计的影响。
- 提出了进程、线程和磁盘管理方案，以反映最新应用的需求。
- 全面讨论常规用途的系统，讲述了实时、嵌入式和超标量构架的概念。
- 强调了关键的评估技术，它们能对操作系统的组件进行有效的比对分析。
- 更全面地讲述了网络概念。
- 增强了对安全主题的讨论，描述了身份验证机制、安全协议、防病毒研究、访问控制方法和无线安全的最新趋势。
- 扩充了对分布式计算的讨论，强调了在 Internet 和万维网领域，它对计算和操作系统的巨大影响。
- 提供了流行的 Intel®构架的细节。
- 提供大量插图、表格、能实际运行的代码示例、伪代码示例和算法。
- 用全新的一章来讲述线程。
- 采用类似于 Java 的语法提供伪代码，便于熟悉 C/C++/Java 的读者理解——几乎所有计算机系的学生都掌握其中的一种或者多种语言。
- 同时使用伪代码和 Java 来讲述多线程主题，演示并发编程中的问题——这使教师能够采用他们喜欢的方式来授课。本版新增了 Java 主题，是可选的。请访问 java.sun.com/j2se/downloads.html 来获取最新版本的 Java。通过下载页面的一个链接可以访问安装指南。
- 加强了对多处理器管理的讲述。
- 新增了线程调度和实时调度主题。
- 包含一个 RAID 讨论。
- 提供了有关 UNIX 进程的一个案例分析。
- 包含有关内容管理和磁盘调度策略的最新信息。
- 在许多章都覆盖了 I/O 系统的重要主题——尤其是第 2 章、第 12 章、第 13 章和最后的 Linux 和 Windows XP 案例分析中(第 20 章和第 21 章)。
- 提供了 730 道自测题及其答案(每一节约 2 道题)，便于学生巩固刚刚学到的知识。

教学方法

本书划分为 8 个部分，每个部分都由一系列相互联系的章构成。包括：

第Ⅰ部分：硬件、软件和操作系统导论
第Ⅱ部分：进程和线程
第Ⅲ部分：物理和虚拟内存
第Ⅳ部分：辅助存储器、文件和数据库
第Ⅴ部分：性能、处理器和多处理器管理
第Ⅵ部分：联网和分布式计算
第Ⅶ部分：安全性
第Ⅷ部分：案例分析
下面描述了本书的教学特点。

引言

每章开始都提供一些引言，有的幽默，有的富有哲理，有的提供有趣的见解。许多读者告诉我们，他们非常喜欢这些与当前章内容相关的引言。也许在读完当前章的内容后，更能体会到引言的寓意。

学习目标

接下来，“学习目标”将列出本章目标，便于您在完成当前章学习之后确定自己是否已经完成这些学习目标。

大纲

本章大纲有助于通过由上而下的方式了解本章内容，所以可以提前知道各章要介绍的概念，并设定一个适合自己的学习进度。

小节和自测题

每一章都通过大量小节来描述重要的操作系统概念。大多数小节都以 2 道自测题及其答案来结束。这些自测题便于你测试自己的知识，获得即时反馈，同时评估你对课本内容的理解。这些自测题还能帮助你为章末的练习题以及随堂考和测验做好准备。有的自测题不能仅仅根据对应小节的内容来回答，解答它们需要学习一些课外的内容。

关键术语

在提供各个术语定义的时候，我们都以中文加英文的形式显示。此外，各章还包含一个“关键术语”小节，其中包含当前章出现的术语及其解释(全书共 1800 个关键术语)。书的最后，还提供了一个按字母排序的完整术语表。“关键术语”小节和“术语表”是一种非常棒的学习工具，有利于复习和参考。

插图

本书包含超过300幅图表、示例和插图，为本书描述的概念提供支持。

Web资源

每章都包含网上资源，通过这些资源，您可以从网上找到更多富有参考价值的研究资源。

小结

每章最后都提供详细的总结性内容，有助于复习当前章介绍的关键概念。

练习题、推荐项目和推荐模拟项目

每一章都包含大量练习题，难度不一。从复习基本的操作系统原理，一直到复杂的推理和研究项目(本书含有900道以上的练习题)。许多OS老师都喜欢布置学期项目，所以我们在每章的练习题之后，包含了“推荐项目”和“推荐模拟项目”小节。

推荐读物

每章都列出推荐阅读的书和文章，并简要回顾对当前章内容而言最重要的资源，以便于你自行进行深入研究。

参考文献

本书需要相当多的研究工作。整本书参考了大量文献(2300个之多)。每个引文在文中都以上标的形式标注，可以在当前章最后的“参考文献”小节找到相应的引文出处。这些引文大部分来自网上。本书前面的版本只使用了标准的书和书面文献，这些书和论文通常很难让读者找到以进行深入研究。现在，可以通过网络直接访问这些文献。此外，对于自己感兴趣的主题，也很容易通过搜索引擎找到相关文章。可以从网上访问到很多学术性期刊，有的是免费的，有的则可以通过专业协会的个人或组织关系访问。网络真不愧为一个资源丰富的“聚宝盆”，从中可以学到很多东西。

索引

我们提供二级索引，以帮助读者迅速找到术语或概念。例如，书中提到的每个操作系统，都按字母顺序出现，并缩进排列在“操作系统”这一术语之下。

补充内容

本书上一版用一整章的篇幅描述了使用队列理论和 Markov 过程进行的分析建模。考虑到操作系统在很大程度上并不属于数学的范畴，所以本版省略了那些内容。相反，操作系统的基础在于“系统思想”(Systems Thinking)。换言之，操作系统主要属于经验主义的范畴。为了更具体地说明这些问题，本版包含了 4 种类型的补充内容，旨在提出挑战、增强趣味性以及拓宽你的知识面。

人物介绍

本书的 18 篇人物小传有助于你了解在操作系统领域有杰出贡献的一些人士，他们是 Edsger Dijkstra，Linus Torvalds，David Cutler，Ken Thompson，Dennis Ritchie，Doug Engelbart，Tim Berners-Lee，Richard Stallman，Gordon Moore，Fernando J. Corbato，Leslie Lamport，Per Brinch Hansen，Peter Denning，Seymour Cray，Bill Gates，Ronald Rivest，Adi Shamir 和 Leonard Adleman。但是，我们对这些人的选择并没有什么特殊的目的，他们只是大量有杰出贡献的人中的一小部分。如果你认为还有其他人应该添加到这个列表上，请务必让我们知道。

小型案例分析

除了针对 Linux 和 Windows XP 操作系统的详尽案例分析之外，还有 14 个针对其他重要操作系统的小型案例分析。这些小型案例分析包括 Mach、CTSS 和 Multics、Unix 系统、实时操作系统、Atlas、IBM 大型机操作系统、VM 操作系统的早期历史、MS-DOS、超级计算机、Symbian OS、OpenBSD、Macintosh、用户模式 Linux(UML)和 OS/2。

奇闻轶事

作为本书作者之一，HMD 先生在学院及业界有数十年的操作系统课程教授经验，他积累了许多奇闻轶事，我们在书中精选了其中的 16 件事情。有的纯属幽默，有的则不然，它们能引出更深层次的哲学问题。希望每一件事都能博你开心一笑。在每一件奇闻轶事的最后，都有一条总结性的“对操作系统设计者的启示”。

操作系统思想

对于操作系统，学院派和实干派的出发点是不尽相同的。学院派醉心于研究操作系统一切理论上的东西，尤其是巧妙的算法、数据结构以及适合进行数学分析的问题。行业专家则必须构造出真正的系统，它们必须满足客户对价格、性能和可靠性的需求。这两个不同的出发点引发了许多有趣的问题。当然，两个派别考虑的问题肯定是既有重复的，也有截然不同的。本书的宗旨是公平探讨学院派和实干派的操作系统理论和实践。

什么是“操作系统思想”？在本书总共43个以“操作系统思想”为题的补充内容中，我们探讨了这个问题。事实上，操作系统的一些问题属于复杂的数学分析的范畴；但是本书作者HMD在计算机行业42年的经验证明，操作系统由于过于复杂，所以不适合在大学4年级或者3年级进行过量的数学分析。即使是研究生，除了一些极窄的领域之外，操作系统不应和数学分析挂上钩。

那么，假如没有一个数学基础来评估操作系统的各个方面，我们如何才能富有成效地考察一个操作系统呢？答案就是我们所谓的“操作系统思想”，本书通过大量题为“操作系统思想”的补充内容来强调与操作系统设计和实现有关的关键概念。

本书包含了下面这些“操作系统思想”：“改革”、“人力和计算机资源的相对价值”、“性能”、“尽量保持简单”、“体系结构”、“高速缓存技术”、“遗留硬件和软件”、“最少权限原则”、“保护”、“试探法”、“用户最终想要的程序”、“操作系统中的数据结构”、“异步与同步”、“并发性”、“并行性”、“标准的一致性”、“可伸缩性”、“信息隐藏”、“等待、死锁和无限延期”、“开销”、“可预测性”、“公平性”、“资源管理强度与相对资源价值的对比”、“对处理能力、内存、存储和带宽的追求是无止境的”、“改变是规则，而不是例外”、“空间资源和碎片化”、“虚拟化”、“经验结果：基于局部性的直观判断”、“缓式分配”、“操作系统中的计算机理论”、“空间-时间折中”、“饱和与瓶颈”、“压缩和解压”、“冗余技术”、“任务关键系统”、“容错”、“加密和解密”、“安全性”、“备份和恢复”、“墨菲法则和健壮系统”、“适当地降低性能”、“数据复制和一致性”、“伦理系统设计”。

案例分析

第20章和第21章深入研究了Linux和Windows XP操作系统。这些全面的案例分析由主要的Linux及Windows XP操作系统开发者进行了审阅。每个案例分析的提纲类似于本书目录，它们相当于对本书关键概念的一个总结——以前各章讲述的是基本原理，而案例分析展示了如何根据这些原理来构建当今最流行的两个操作系统。Linux案例分析以最新的kernel release(v. 2.6)为蓝本，Windows XP案例分析则反映了最新的Windows操作系统特性。

本书导读

本书分为8部分，共21章。下面将对全书内容进行概述。

第I部分“硬件、软件和操作系统导论”包含两章，介绍操作系统的概念、操作系统发展简史、一些基本的硬件和软件概念，为读者后续学习奠定基础。

第1章“操作系统导论”，定义“操作系统”这个术语，并解释为什么需要此类系统。本章从历史角度探讨操作系统，追溯它们在20世纪后半期近50年的发展历程。在这一章，我们审视20世纪50年代的批处理系统，讲到20世纪60年代并行处理发展趋势和多道程序设计的萌芽——同时讲到批处理的多道程序设计和交互式分时系统。我们阐述一些重要的操作系统的发展历程，包括CTSS、Multics、CP/CMS和UNIX。本章提出这样一种思想：

在计算资源成本远远高于人力资源成本那个年代，操作系统设计师所做的工作是很了不起的(时至今日，他们仍然是了不起的)。我们追溯 20 世纪 70 年代计算机联网技术、Internet 和 TCP/IP 协议集的演变，了解到个人计算革命的起源。在 20 世纪 80 年代，IBM 个人电脑和 Apple Macintosh(后者因为有图形用户界面而得以流行)的发布，标志着个人计算的成熟。本章还介绍分布式计算的起源和客户端/服务器模型的发展。在 20 世纪 90 年代，Internet 的使用随着万维网的普及呈爆炸性增长。微软成为业界数一数二的软件厂商，并发布了 Windows NT 操作系统(今天 Windows XP 操作系统的前身，本书第 21 章将对这一主题进行阐述)。随着 C++和 Java 等编程语言的流行，对象技术成为主流的开发模式。开源软件的迅速增长促进了 Linux 操作系统的巨大成功(本书第 20 章将对这一主题进行阐述)。我们讨论了操作系统如何为应用程序的开发提供平台。嵌入式系统也在我们的讨论之列，主要强调在对可靠性要求较高的任务关键和业务关键系统中嵌入式系统的重要性。我们讨论了操作系统的各个组成部分和操作系统的重要目标(本书将坚持从各个方面关注操作系统的性能问题)。操作系统的体系结构也在本章略作介绍，包括单内核体系结构、分层体系结构、微内核体系结构和网络及分布式操作系统。

第 2 章“硬件和软件概念”，总结操作系统管理的硬件和软件资源。本章介绍硬件设计趋势——处理能力、存储容量和通信带宽显著增长影响着操作系统的设计，反之亦然。本章讨论各种各样的硬件组件，包括主板、处理器、时钟、主存储器、辅助存储器、总线、直接存取内存(DMA)、外围设备等。我们介绍最近或最新的硬件技术，讨论操作系统支持的硬件，包括处理器执行模型、特权指令、定时器、时钟、自举和即插即用等。本章还讨论缓存和缓冲区等性能增强技术。此外还将分析编译、链接、加载、机器语言、汇编语言、解释器、编译器、高级语言、结构化编程、面向对象编程和应用程序编程接口(API)等软件概念。本章还介绍了固件和中间件技术。

第 II 部分“进程和线程”包含 6 章内容，介绍以下概念：进程、线程、进程和线程状态转换、中断、上下文切换、异步、互斥、监视器、死锁、延迟、进程和线程的处理器调度等。

第 3 章“进程的概念”，首先通过定义进程的基本概念来讨论操作系统基础。我们考虑了进程在两个进程状态之间转换的生命周期。讨论了进程的进程控制块或进程描述符表示，重点介绍数据结构在操作系统中的重要作用。本章还讨论操作系统为什么需要进程结构，描述可以对进程执行的操作，比如挂起和恢复。多道程序设计注意事项、执行挂起的进程和上下文切换也在本章讨论之列。本章还讨论了中断。要成功实现任何一个多道程序设计环境，中断是最关键的因素之一。我们分析了中断处理、中断类型、信号和消息传输的进程间通信。本章最后将提供一个 UNIX 进程案例分析。

第 4 章“线程的概念”，把进程概念的讨论延伸到并发程序执行的一个更小的单元：线程。本章定义了何为线程，解释了它们与进程的关系。线程的生命周期以及线程如何在不同状态之间转换也在本章讨论之列。本章讨论各种线程结构模型，其中包括用户级线程、内核级线程以及用户级-内核级混合线程。本章介绍实现线程的注意事项，包括线程信号传输和线程中止。我们讨论了 POSIX 标准及其线程规范 Pthreads。本章最后介绍 Linux、Windows XP 和 Java 线程实现机制。在介绍 Java 线程的时候，我们提供了一个完整的、可运行的 Java 程序，并附有输出结果(注意，本书用到的所有 Java 程序的源代码均可以从 www.deitel.com.cn/ose3 下载)。

第 5 章“异步并发执行”，讨论多道程序设计系统中遇到的并发问题。本章介绍互斥问题，以及线程如何对共享资源的访问进行管理。本章的一大特色是 Java 多线程案例分析：Java 中的生产者/消费者关系。该案例使用一个完整的、可运行的 Java 程序和几个输出结果，清楚地展示在并发线程访问共享资源时，如果不采用同步机制，会发生什么情况。这个例子清楚地表明，此类并发程序在有些情况下，能够正常运行，但在有些情况下，会产生错误结果。在第 6 章，我们将展示如何用 Java 多线程程序来解决这个问题。程序代码中的“临界区”(critical section)的概念也在本章介绍。针对临界区访问保护，本章介绍了几种软件机制，并提供了互斥问题的解决办法，其中包括 Dekker 算法、Peterson 算法和 Lamport 的 Bakery 算法的 N 线程互斥机制。本章还讨论了有助于实现互斥算法的硬件机制，包括禁止中断、test-and-set 指令和 swap 指令。最后介绍信号量(用于实现互斥和线程同步的高级算法)，二元信号量和计数信号量均有涉及。

第 6 章“并发编程”，首先介绍监视器(一种高级的互斥结构)的概念，进而过渡到解决并发编程过程中的几个经典问题，首先用伪码监视器，然后是一个附有输出结果且完整的、可运行的 Java 程序。监视器是以类 C/C++/Java 伪码的风格给出的。我们解释了监视器如何实施信息隐藏，讨论了监视器条件变量如何不同于约定的变量。本章阐述如何用一个简单的监视器来控制对互斥资源的访问。然后讨论并发编程的两个经典问题：循环缓冲区和 reader/writer 问题。我们用伪代码监视器实现了每个问题的解决方案，解释了 Java 监视器，并探讨了 Java 监视器和其他经典监视器的区别。本章继续之前的 Java 多线程案例分析，这一次用 Java 实现了一个生产者/消费者关系和一个循环缓冲区。读者可以基于这个程序，用 Java 来实现 reader/writer 问题的一个解决方案。

第 7 章“死锁和无限延期”介绍等待事件响应的两个可能的严重后果：死锁和无限延期。我们主要关心的是，管理等待实体的系统必须进行精心设计，以避免此类问题。本章介绍了死锁的几个例子，其中包括交通死锁、经典的单行道大桥、一个简单的资源死锁、脱机打印系统中的死锁和 Dijkstra 的有趣的哲学家就餐问题。本章介绍关键的资源概念，其中包括抢占、共享、可重入和可连用。本章正式定义了死锁并讨论了如何进行死锁的预防、避免、检测和恢复(基本上都不轻松)。死锁的 4 个必要条件也在本章讨论之列，分别为互斥、等待、不可抢占和循环等待。我们解释了 Havender 的方法，通过杜绝其余 3 个条件中的任何一个，可以避免死锁。相较于预防死锁，避免死锁能更灵活地分配资源。本章解释了 Dijkstra 的银行家算法如何用于避免死锁，展示了安全状态例子、非安全状态例子和安全状态-非安全状态转换例子。此外，还讨论了该算法的不足之处。我们还介绍了如何用缩减资源分配图来检测死锁。最后讨论了当前和未来系统中的死锁策略。

第 8 章“处理器调度”针对为进程和线程分配处理器时间，讨论了相关的一些概念和算法。调度级别、目标和标准也在讨论之列。我们比较了可抢占和不可抢占调度方法。解释了如何在调度算法中精心设置优先级和时间片(quanta，处理器时间的有限分配时间)。本章还介绍了几个经典和现代的调度算法，其中包括先入先出(first-in-first-out，FIFO)、轮转(round robin，RR)、最短进程优先(shortest-process-first，SPF)、最高响应比优先(highest-response-ratio-next，HRRN)、最短剩余时间优先(shortest-remaining-time，SRT)、多级反馈队列和公平共享调度。每个算法都是用吞吐量、平均响应时间和响应时间的各种变化形式等单位来计算的。我们讨论了 Java 线程调度、软实时调度、硬实时调度和截止期调度。

第III部分“物理和虚拟内存”包含3章内容，讨论了物理内存和虚拟内存系统中的内存组织和管理。

第9章“实内存组织和管理”从历史角度讨论操作系统是如何组织和管理实内存资源的。我们讨论了从简单的到复杂的方案，最终目的是找到一种最佳方式来使用宝贵的主存资源。本章回顾了内存的层次结构，内存由高速缓存、主内存和辅助存储构成。然后，我们讨论3种内存管理策略，分别为取页(fetch)、安置和替换。连续性内存分配和非连续性内存分配方案也在本章讨论之列。在讲到单用户连续性内存分配的时候，我们讨论了叠加、保护和单流批处理。我们追溯了多道程序设计内存组织的发展历程，从固定分区的多道程序设计到可变分区的多道程序设计，涉及的问题有内部和外部内存碎片，以及为减少内存碎片而采用的内存压缩和合并技术。本章讨论了最先适应、最佳适应和最坏适应内存放置策略，最后讨论了使用内存交换的多道程序设计。

第10章“虚拟内存组织”描述虚拟内存的基本概念和支持虚拟内存的硬件功能。本章描述了虚拟内存的必要性，讨论了它的典型的实现方式。对虚拟内存进行组织的主要方法是分页和分段，本章对它们各自的优缺点进行了分析。在讨论分页系统时，我们重点讨论了如何通过直接映射来实现分页地址转换，如何通过关联式映射来实现分页地址转换，如何通过直接/关联式映射来实现分页地址转换，其中，介绍了多级页表、反向页表以及分页系统中的共享技术。在讨论分段系统时，我们重点讨论了如何通过直接映射来实现分段地址转换、分段系统中的共享以及分段系统中的保护和访问控制。我们还讨论了混合式的分段/分页系统，讨论了动态地址转换，以及在这种系统中的共享和保护。本章最后探讨了流行的IA-32 Intel构架的虚拟内存实现。

第11章“虚拟内存管理”继续讨论虚拟内存，本章分析了操作系统如何对虚拟内存的性能进行优化。由于分页系统目前占据统治地位，所以本章将重点放在页管理上，尤其是页替换策略。本章讨论了操作系统领域最重要的研究成果之一，也就是局部性思想。我们从时间和空间的角度对它进行了分析。我们讨论了页应该在什么时候进入内存，研究了请求分页和先行分页技术。在可用内存缺少的时候，进入的页必须替换内存中现有的页——操作系统的页替换策略可能对性能产生显著影响。本章探讨了许多页替换策略，其中包括随机、先入先出(FIFO 和 Belady 异常)、最近最少使用(LRU)、最少使用(LFU)、最近不用(NUR)、二次机会、时钟和分段队列(SEGQ)、最远的页和页错误频率(PFF)。本章还讨论了Denning 的经典程序行为工作集模型，以及各种页替换策略如何达到它的目标。讨论了自动页释放的可能性。探讨了人们对于小页和大页的争论，并描述了分页时的程序行为。本章最后讨论了Linux页替换的例子，并对全局与局部页替换策略进行了对比。

第IV部分“辅助存储器、文件和数据库”用两章的篇幅描述了操作系统对辅助存储器上的数据进行管理的技术。我们讨论了硬盘性能优化以及文件和数据库系统。贯穿全书，我们从多个角度讲解了 I/O 系统，尤其是在第 2 章、第 12 章以及本书最后的 Linux 和Windows XP 案例分析中(分别是第20章和第21章)。

第12章“磁盘性能优化”将重点放在活动臂硬盘存储器的特征上，并解释了操作系统如何优化它的性能。本章讲述了辅助存储设备的演变和活动臂磁盘存储器的特征。在半个世纪的爆炸性技术进步中，活动臂磁盘存储设备也在不断地改良自己，以适应时代的进步。我们解释了磁盘调度的必要性，并演示了如何从活动臂磁盘硬盘设备中获得高性能。

我们描述了磁盘调度策略的不断发展，其中包括寻道优化策略，在这个策略中，介绍了先到先服务(FCFS)、最短寻道时间优先(SSTF)、SCAN、C-SCAN、FSCAN、N-Step SCAN、LOOK、C-LOOK 和 VSCAN(在习题中解释)。还描述了旋转优化策略，其中包括 SLTF、SPTF 以及 SATF。我们还讨论了其他流行的磁盘系统性能技术，其中包括高速缓存、缓冲、碎片整理、数据压缩和记录块技术。本书这一版全面地讲述了 RAID 系统。RAID(独立磁盘冗余阵列)是一系列技术的统称，它们为普通的磁盘系统赋予了更高的性能和容错性。本章描述了各个关键的 RAID“级别”，其中包括第 0 级(带区集)、第 1 级(镜像)、第 2 级(位级汉明 ECC 奇偶校验)、第 3 级(位级 XOR ECC 奇偶校验)、第 4 级(块级 XOR ECC 奇偶校验)和第 5 级(块级分布式 XOR ECC 奇偶校验)。

第 13 章“文件和数据库系统”讨论了操作系统如何组织和管理文件和数据库。我们解释了一些关键性的概念，其中包括数据层次结构、文件、文件系统、目录、链接、元数据和安装。本章描述了各种文件组织方法，其中包括顺序、直接、索引顺序和分区文件组织。我们还探讨了各种文件分配方法，其中包括连续文件分配、链表式非连续文件分配、表格式非连续文件分配以及索引非连续文件分配。我们解释了如何实现用户级访问控制以及访问控制矩阵。讨论了基本访问方法、排队访问方法、先行缓冲以及内存映射文件等数据访问技术。解释了用于确保数据完整性的技术，其中包括保护、备份、恢复、日志、原子事务处理、回滚、提交、检查点、阴影页和使用日志构造的文件系统。本章谈及了文件服务器和分布式系统，并指引读者到第 16 章～第 18 章去更全面地了解这些主题。本章还介绍了数据库系统，分析了它们的优势。另外，我们还讨论了用于支持数据库系统的数据访问、关系数据库模型和操作系统服务。

第Ⅴ部分“性能、处理器和多处理器管理”用两章的篇幅讨论了性能监视、测量和评估技术，并强调了使用多处理器系统能获得的强大性能。

第 14 章“性能和处理器设计”重点讨论了操作系统设计者最重要的设计目标之一，即确保系统的高性能。另外，还讨论了为了满足性能要求，特定类型的处理器所扮演的重要角色。本章归纳了性能指标，讨论了绝对性能指标、易用性、周转时间、响应时间、系统反应时间、响应时间方差、吞吐量、工作负荷、容量和利用率等问题。我们讨论了各种性能评估技术，其中包括 tracing 和 profiling、计时、微基准、特定应用的评估、分析模型、基准程序、复合程序、仿真和性能监视等。瓶颈和饱和问题在这章得到了讨论。我们演示了系统如何动态调整到正反馈和负反馈。一个系统的性能在很大程度上依赖于其处理器的性能，而处理器的性能又受到其指令集的严重影响。本章探讨了核心构架，其中包括复杂指令集计算(CISC)、精简指令集计算(RISC)和各种 Post-RISC 处理器。本章最后讨论了显式并行指令计算(EPIC)。

第 15 章“多处理器管理”深入讨论了多处理器的硬件和软件问题。为了构建更强大的计算系统，一个办法是采用大量处理器。本章首先给出了一个关于超级计算机的小型案例分析，然后是超级计算之父 Seymour Cray 的一个人物传记。我们描述了多处理器构架，讨论了顺序和并行构架分类、处理器互连方案以及松散和紧密构架的对比。我们讨论了多处理器操作系统组织方案，其中包括主/从、独立内核以及对称结构。我们解释了内存访问构架，包括均匀内存访问(UMA)、非均匀内存访问(NUMA)、全高速缓存存储构架(COMA)以及非远程存储访问(NORMA)。本章讨论了多处理器内存共享问题，描述了缓存一致性、

页复制和迁移以及共享虚拟内存等。在本章，我们将继续讨论始于第 8 章的处理器调度问题，描述了不知道作业(job blind)和知道作业(job aware)多处理器调度。本章还讨论了进程迁移和进程迁移步骤的问题，并研究了静态和动态负载平衡策略。本章解释了如何通过旋转锁(spin lock)、休眠/唤醒锁和读/写锁来强制多处理器互斥。

第Ⅵ部分“联网和分布式计算”用 3 章的篇幅讲述了网络、联网系统和分布式系统。

第 16 章“联网概论”描述了计算机联网的基础知识。随后两章在讲述分布式计算的主题时，会以本章的内容为基础。本章讨论了网络拓扑，其中包括总线、环形、网状、完全连接网状、星形和树形拓扑。此外，我们还解释了无线网络所带来的挑战。我们讨论了局域网(LAN)和广域网(WAN)，并对 TCP/IP 协议堆栈进行了全面的讨论。本章讨论了包括超文本传输协议(HTTP)和文件传输协议(FTP)在内的应用层协议，还讨论了包括传输控制协议(TCP)和用户数据文报协议(UDP)在内的传输层协议。在网络层上，本章探讨了 Internet 协议(IP)以及它的最新版本 IPv6。本章还讨论了一些链路层协议，包括 Ethernet、令牌环、光纤分布式数据接口(FDDI)和 IEEE 802.11(无线)。最后讨论了客户/服务器模型和 n 层系统。

第 17 章“分布式系统入门”介绍了分布式操作系统，并讨论了分布式系统的特点，其中包括性能、伸缩性、连接性、安全性、可靠性、容错性和透明性。本章对网络操作系统和分布式操作系统进行了比较。讨论了分布式系统中的通信以及“中间件”技术所扮演的关键角色，其中包括远程过程调用(RPC)、远程方法调用(RMI)、公共对象请求中介构架(CORBA)以及 Microsoft DCOM(分布式组件对象模型)。本章讨论了分布式系统中的进程迁移、同步和互斥，讨论了没有共享内存的互斥，以及 Agrawala 和 Ricart 的分布式互斥算法。也讨论了分布式系统中的死锁，重点讲述了死锁预防和死锁检测。在本章最后，我们提供了 Sprite 和 Amoeba 分布式系统的案例分析。

第 18 章“分布式系统和 Web 服务”继续讨论分布式系统，重点讲述了分布式文件系统、群集、对等计算、网格计算、Java 分布式计算以及 Web 服务。本章首先探讨了分布式系统的一些特点和注意事项，其中包括透明性、伸缩性、安全性、容错性和一致性。我们还就当前的一些关键的分布式文件系统给出了案例分析，其中包括网络文件系统(NFS)、Andrew 文件系统(AFS)、Coda 文件系统和 Sprite 文件系统。我们讨论了群集处理，针对高性能群集、高可用性群集以及负载平衡群集进行了详细的探讨。我们研究了多个群集的例子，其中包括基于 Linux 的 Beowulf 群集，以及基于 Windows 的群集。我们探讨了对等(P2P)分布式计算，研究了 P2P 和客户机/服务器应用程序之间的关系，对集中式和分散式 P2P 应用程序进行了对比，并讨论了 Peer 发现与搜索问题。我们研究了 Sun Microsystems 公司的 JXTA 项目，描述了 JXTA 如何创建一个用于构建 P2P 应用程序的框架。随后，本章开始讨论网格计算，指出它如何利用分布于全球的个人和商业电脑的闲散运算能力来执行大规模计算，从而解决一些复杂的计算问题。我们讨论了关键的 Java 分布式计算技术，其中包括 Java servlet 和 JavaServer Page(JSP)、Jini、JavaSpaces 和 Java Management Extensions(JMX)。本章最后探讨了 Web 服务，并介绍了两种关键的 Web 服务平台——Microsoft 的.NET 平台和 Sun Microsystems 的 Sun ONE 平台。

第Ⅶ部分“安全性”由一章构成。

第 19 章“安全性”讨论了计算机和网络安全技术，操作系统可以利用这些技术来提供一个安全的计算环境。本章讨论了密钥和公钥加密技术，以及流行的 RSA 和 PGP 公钥

算法。探讨了身份验证问题，其中包括密码保护、密码加码、生物特征、智能卡、Kerberos和单一登录等。我们讨论了访问控制，探讨了访问权限、保护域、访问控制模型、访问控制策略和访问控制机制。在访问控制机制中，我们讨论的主题包括访问控制矩阵、访问控制列表和权能列表。本章对各种安全攻击进行了描述，其中包括密码分析、病毒、蠕虫、拒绝服务(DoS)攻击、利用软件漏洞以及系统渗透等。我们讨论了各种攻击预防和安全方案，其中包括防火墙和入侵检测系统(IDS)、防病毒软件、安全补丁和安全文件系统等。并介绍了美国联邦政府橘皮书(Orange Book)安全性分类系统，同时提供了关于 OpenBSD 的一个小型案例分析。目前，许多人都认为 OpenBSD 是最安全的操作系统。本章也对安全通信进行了探讨，并描述了一次成功的、安全的事务处理必须符合什么要求——保密性、完整性、身份验证、授权和不可抵赖。介绍了密钥协定协议，它指示两个团体通过非安全媒介交换秘密密钥的过程。我们解释了数字签名，还深入探讨了公钥基础设施(PKI)，其中包括数字证书和证书颁发机构。本章讨论了各种安全通信协议，其中包括 SSL、VPN、IPSec以及无线安全性。讨论了隐写术，也就是如何在信息中隐藏别的信息。这样可以将秘密的消息隐藏在公开传输的消息中。另外，还可以利用这种技术来保护知识产权，比如利用数字化水印。本章比较了专利性和开源安全方案，并在最后提供了一个 UNIX 安全性的案例分析。

第Ⅷ部分“操作系统案例分析”由两章构成，对 Linux 2.6 内核和 Microsoft Windows XP 操作系统进行了深入的技术研究。每个案例分析都按照本书的大纲来编排，从而有助于读者理解本书之前的各个主题。

第 20 章“案例分析：Linux”深入探讨了 Linux 2.6。本章讨论了 Linux 的发展史、开发团体及世界上最流行的开源操作系统分发。考察了 Linux 操作系统中的核心组件，特别分析和解释了对应前些章中概念的实现，包括内核体系结构、进程管理(进程、线程和调度)、内存组织和管理、文件系统(虚拟文件系统、虚拟文件系统高速缓存、ext2fs 和 procfs)、I/O管理(设备驱动程序、字符设备 I/O、块设备 I/O、网络设备 I/O、统一设备模型和中断)、同步(自旋锁、阅读程序/写程序锁、顺序锁和内核信号量)、IPC(信号、管道、套接字、消息队列、共享内存和系统 V 信号量)、联网技术(数据包处理、网络过滤器框架和 hooks)、可伸缩性(对称多处理、非均衡内存访问及嵌入式 Linux)和安全性。

第 21 章“案例分析：Windows XP”通过对当今最流行的商业化操作系统 Windows XP 的全面剖析，对上一个 Linux 案例分析的内容进行了有益的补充。这个案例分析考察了 Windows XP 操作系统中的核心组件，探讨了它们如何交互地向用户提供服务。介绍了 Windows 的发展史，包括 Bill Gates 和 David Cutler 的人物介绍。介绍了 Windows XP 的设计目标及其体系结构，同时研究了硬件抽象层(HAL)、微内核、执行组件、环境子系统、动态链接库(DLL)和系统服务等关键技术。详细讨论了系统管理机制，包括注册表、对象管理器、中断请求级(IRQL)、异步过程调用(APC)、延迟过程调用(DPC)和系统线程。我们还研究了进程和线程管理，包括控制块、线程局部存储(TLS)、创建和终止进程、作业、纤程和线程池。关于线程的调度介绍了线程状态、线程调度算法、判断线程优先级和多处理器调度，研究了线程同步，考察了调度程序对象、事件对象、互斥对象、信号量对象、可等待计时器对象、内核模式锁和其他同步工具。解释了内存管理、内存组织、内存分配和页替换问题。探讨了文件系统管理，关于文件系统驱动程序和 NTFS 主题介绍了主文件表、

数据流、文件压缩、文件加密、稀疏文件、重新分析点和安装卷。讨论了 I/O 管理，包括设备驱动程序、即插即用、电源管理、Windows 驱动程序模型(WDM)、I/O 处理、I/O 请求包、同步 I/O、异步 I/O、数据传输技术、中断处理和文件高速缓存管理。关于进程间通信机制介绍了管道(匿名和命名)、mailslot、共享内存及本地和远程过程调用。介绍了 Windows 的组件对象模型(COM)和拖放式及复合文档。关于联网技术介绍了网络 I/O、网络驱动程序体系结构、网络协议(TCP/IP 之上的 IPX、SPX、NetBEUI、NetBIOS、WinHTTP、WinINet 和 Winsock 2)。网络服务介绍了活动目录、LDAP 和远程访问服务。介绍了替代 DCOM 的.NET 技术。讨论了 Windows 的可伸缩性问题，及其对对称多处理(SMP)和嵌入式 Windows XP 的支持。在本章的最后，讨论了安全性问题，包括授权、认证、Internet 连接防火墙(ICF)和其他安全特性。

本书的教辅资料

本书为教员准备了丰富的教辅资料。在面向教员的 CD 上包含了每章后面大部分的练习题答案。教员可通过 Prentice Hall 代表使用这张 CD。请访问 vig.prenhall.com/replocator 查找您的 Prentice Hall 代表。[注意：请不要管我们索要这张 CD。该 CD 的数量非常有限，只提供给大学教授用于本书配套的教学，教员只能通过 Prentice Hall 代表得到习题答案。]本书的教辅资料还包括一个多项选择题形式的测试。此外，我们还提供一个 PowerPoint 演示文稿，其中包含了本书中所有的图、表、示例、代码和关键内容的总结。教员可以定制这个演示文稿，可从 www.deitel.com 下载，在 Prentice Hall 本书的站点(www.prenhall.com/deitel)上也包含了这个演示文稿。该站点同时还为教员提供了一个课程大纲管理器，它有助于教员规划互动课程及创建在线课程。

同样，该配套网站还为学生准备了许多有用的资料：

- 可定制的 PowerPoint 演示文稿。
- Java 程序的源代码

为学生提供的各章的可用资源包括：

- 章学习目标
- 大纲
- 章小结
- Web 链接

致谢

写一本教科书最高兴的事情之一，就是感谢许多姓名没有出现在封面上的人士，他们的辛苦工作、合作、友谊以及理解，是本书得以顺利出版的关键。Deitel & Associates 的许多同事在本书及其教辅上面花费了大量宝贵的时间，其中包括：Jeff Listfield，Su Zhang，Abbey Deitel，Barbara Deitel 和 Christi Kelsey。

这里要特别感谢 Ben Wiedermann 对本书第二版材料的审阅工作。他审阅和更新了现有的大量文字和图表。他还组织并指挥着一个由 Deitel 实习学生构成的团队来研究操作系统的最新进展，并将它们包括到新的一版当中。Ben 目前正在奥斯汀的德州大学攻读计算机科学博士。

还要感谢一些 Deitel & Associates 大学实习活动的参与者，他们在本书及其教辅包上付出了大量辛勤的工作：Jonathan Goldstein(Cornell 大学)，Lucas Ballard(Johns Hopkins 大学)，Tim Christensen(Boston 大学)，John Paul Casiello(Northeastern 大学)，Mira Meyerovich(Brown 大学博士研究生)，Andrew Yang(Carnegie Mellon 大学)，Kathryn Ruigh(Boston 大学)，Chris Gould(Massachusetts 大学 Lowell 分校)，Marc Manara(Harvard 大学)，Jim O'Leary(Rensselaer Polytechnic 学院)，Gene Pang(Cornell 大学)以及 Randy Xu(Harvard 大学)。

还要感谢 Howard Berkowitz(以前就职于 Open System International 公司)参与创作本书第二版的第 16 章“分布式计算”。本书第三版的第 16 章到第 18 章对联网和分布式计算进行了更全面的、更深入的以及更符合当前形势的讲述，其基础正是 Howard Berkowitz 在本书第二版的工作。

和 Prentice Hall 出版社的专家共事，我们感觉非常幸运。尤其要感谢我们的计算机科学编辑 Kate Hargett 以及她的老板兼我们的出版导师——Prentice-Hall 工程和计算机科学部门主编 Marcia Horton。Vince O'Brien，Tom Manshreck，John Lovell 和 Chirag Thakkar 出色地管理着本书的生产过程。Sarah Parker 负责管理本书教辅包的出版工作。

还要感谢为本书(指英文原书)赋予了一个全新的外观和感觉的设计团队：Carole Anson，Paul Belfanti，Geoffrey Cassar 和 John Root。还要感谢 Artworks 的富有创造力的团队，其中包括 Kathryn Anderson，Jay McElroy，Ronda Whitson，Patricia Burns，Matt Haas，Xiaohong Zhu，Dan Missildine，Chad Baker，Sean Hogan，Audrey Simonetti，Mark Landis，Royce Copenheaver，Stacy Smith，Scott Wieber，Pam Taylor，Anna Whalen，Cachy，Shelly，Ken Mooney，Tim Nguyen，Carl Smith，Jo Thompson，Helenita Zeigler 和 Russ Crenshaw。

感谢在各章的“推荐读物”和“参考文献”中提到的很多位作者，他们的研究论文、文章和书籍对这本《操作系统》提供了补充，也指引读者步入了广阔的操作系统领域。

非常感谢本书第三版的审稿人，尤其要感谢的是 Prentice Hall 出版社的 Carole Snyder 和 Jennifer Cappello，她们负责对本书的审稿进行统筹。在时间紧迫的情况下，这些审稿人对书稿进行了细致的检查，提供了大量建议来提高书稿的准确性和完整性。能得到这些聪明和繁忙的专家们的指导，使我们受益无穷。

第三版的审稿人：

TigranAivazian,Veritas Software, Ltd.
Jens Axboe, SUSE Labs
Dibyendu Baksi, Scientific Technologies Corporation
Columbus Brown, IBM
Fabian Bustamante, Northwestern University
Brian Catlin,Azius Developer Training
Stuart Cedrone,Hewlett-Packard