

教育部高等教育司推荐
国外优秀信息科学与技术系列教学用书

操作系统概念

(第七版 影印版)

OPERATING SYSTEM CONCEPTS
(Seventh Edition)

■ Abraham Silberschatz
Peter Baer Galvin
Greg Gagne



WILEY



高等教育出版社
Higher Education Press

教育部高等教育司推荐
国外优秀信息科学与技术系列教学用书

TP316

Y51

2007.

操作系统概念

(第七版 影印版)

OPERATING SYSTEM CONCEPTS

(Seventh Edition)

Abraham Silberschatz

Peter Baer Galvin

Greg Gagne

江苏工业学院图书馆

藏书章



高等教育出版社

Higher Education Press

图字：01-2006-5426 号

Operating System Concepts, Seventh Edition

Abraham Silberschatz, Peter Baer Galvin, Greg Gagne

本书封面贴有 John Wiley & Sons, Inc. 防伪标签, 无标签者不得销售。

Copyright ©2005 John Wiley & Sons, Inc.

All Rights Reserved.

**AUTHORIZED REPRINT OF THE EDITION PUBLISHED BY JOHN WILEY & SONS, INC.,
NEW YORK.,** New York, Chichester, Weinheim, Singapore, Brisbane, Toronto. No Part of this book may
be reproduced in any form without the written permission of John Wiley & Sons, Inc.

For sale and distribution in the People's Republic of China exclusively (except Taiwan, Hong Kong
SAR and Macau SAR).

仅限于中华人民共和国境内(但不允许在中国香港、澳门特别行政区和中国台湾地区)销售发行。

图书在版编目(CIP)数据

操作系统概念 = Operating System Concepts: 第7版:

英文 / (美)西尔伯查茨(Silberschatz, A.), (美)

高尔文(Galvin, P.B.), (美)加根(Gagne, G)著.

影印本. —北京: 高等教育出版社, 2007.3

ISBN 978-7-04-020928-0

I. 操… II. ①西… ②高… ③加… III. 操作系统—高等
学校—教材—英文 IV. TP316

中国版本图书馆 CIP 数据核字(2007)第 027871 号

策划编辑 武林晓 责任编辑 康兆华

封面设计 张楠 责任印制 陈伟光

出版发行 高等教育出版社

社址 北京市西城区德外大街4号

邮政编码 100011

总机 010-58581000

经销 蓝色畅想图书发行有限公司

印刷 北京民族印刷厂

开本 787×1092 1/16

印张 59.5

字数 1 190 000

购书热线 010-58581118

免费咨询 800-810-0598

网址 <http://www.hep.edu.cn>

<http://www.hep.com.cn>

网上订购 <http://www.landaco.com>

<http://www.landaco.com.cn>

畅想教育 <http://www.widedu.com>

版次 2007年3月第1版

印次 2007年3月第1次印刷

定价 65.00元

本书如有缺页、倒页、脱页等质量问题, 请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 20928-00

序

20 世纪末, 以计算机和通信技术为代表的信息科学和技术对世界经济、科技、军事、教育和文化等产生了深刻影响。信息科学技术的迅速普及和应用, 带动了世界范围信息产业的蓬勃发展, 为许多国家带来了丰厚的回报。

进入 21 世纪, 尤其随着我国加入 WTO, 信息产业的国际竞争将更加激烈。我国信息产业虽然在 20 世纪末取得了迅猛发展, 但与发达国家相比, 甚至与印度、爱尔兰等国家相比, 还有很大差距。国家信息化的发展速度和信息产业的国际竞争能力, 最终都将取决于信息科学技术人才的质量和数量。引进国外信息科学和技术优秀教材, 在有条件的学校推动开展英语授课或双语教学, 是教育部为加快培养大批高质量的信息技术人才采取的一项重要举措。

为此, 教育部要求由高等教育出版社首先开展信息科学和技术教材的引进试点工作。同时提出了两点要求, 一是要高水平, 二是要低价格。在高等教育出版社和信息科学技术引进教材专家组的努力下, 经过比较短的时间, 第一批引进的 20 多种教材已经陆续出版。这套教材出版后受到了广泛的好评, 其中有不少是世界信息科学技术领域著名专家、教授的经典之作和反映信息科学技术最新进展的优秀作品, 代表了目前世界信息科学技术教育的一流水平, 而且价格也是最优惠的, 与国内同类自编教材相当。

这项教材引进工作是在教育部高等教育司和高教社的共同组织下, 由国内信息科学技术领域的专家、教授广泛参与, 在对大量国外教材进行多次遴选的基础上, 参考了国内和国外著名大学相关专业的课程设置进行系统引进的。其中, John Wiley 公司出版的贝尔实验室信息科学研究中心副总裁 Silberschatz 教授的经典著作《操作系统概念》, 是我们经过反复谈判, 做了很多努力才得以引进的。William Stallings 先生曾编写了在美国深受欢迎的信息科学技术系列教材, 其中有多种教材获得过美国教材和学术著作者协会颁发的计算机科学与工程教材奖, 这批引进教材中就有他的两本著作。留美中国学者 Jiawei Han 先生的《数据挖掘》是该领域中具有里程碑意义的著作。由达特茅斯学院 Thomas Cormen 和麻省理工学院、哥伦比亚大学的几

位学者共同编著的经典著作《算法导论》，在经历了 11 年的锤炼之后于 2001 年出版了第二版。目前任教于美国 Massachusetts 大学的 James Kurose 教授，曾在美国三所高校先后 10 次获得杰出教师或杰出教学奖，由他主编的《计算机网络》出版后，以其体系新颖、内容先进而倍受欢迎。在努力降低引进教材售价方面，高等教育出版社做了大量和细致的工作。这套引进的教材体现了权威性、系统性、先进性和经济性等特点。

教育部也希望国内和国外的出版商积极参与此项工作，共同促进中国信息技术教育和信息产业的发展。我们在与外商的谈判工作中，不仅要坚定不移地引进国外最优秀的教材，而且还要千方百计地将版权转让费降下来，要让引进教材的价格与国内自编教材相当，让广大教师和学生负担得起。中国的教育市场巨大，外国出版公司和国内出版社要通过扩大发行数量取得效益。

在引进教材的同时，我们还应做好消化吸收，注意学习国外先进的教学思想和教学方法，提高自编教材的水平，使我们的教学和教材在内容体系上，在理论与实践的结合上，在培养学生的动手能力上能有较大的突破和创新。

目前，教育部正在全国 35 所高校推动示范性软件学院的建设和实施，这也是加快培养信息科学技术人才的重要举措之一。示范性软件学院要立足于培养具有国际竞争力的实用性软件人才，与国外知名高校或著名企业合作办学，以国内外著名 IT 企业为实践教学基地，聘请国内外知名教授和软件专家授课，还要率先使用引进教材开展教学。

我们希望通过这些举措，能在较短的时间，为我国培养一大批高质量的信息技术人才，提高我国软件人才的国际竞争力，促进我国信息产业的快速发展，加快推动国家信息化进程，进而带动整个国民经济的跨越式发展。

教育部高等教育司

二〇〇二年三月

To my children, Lemor, Sivan, and Aaron

Avi Silberschatz

*To my wife, Carla,
and my children, Gwen Owen and Maddie*

Peter Baer Galvin

*In memory of Uncle Sonny,
Robert Jon Heileman 1933 — 2004*

Greg Gagne

前 言

操作系统是计算机系统的基本组成部分。同样,“操作系统”课程也是计算机科学教育的基本组成部分。随着计算机在众多领域(从儿童游戏到极为尖端的、用于政府和多国企业进行规划的工具等)得到广泛应用,操作系统也正在以惊人的速度发生着变化。然而,操作系统的基本概念仍然是比较清晰的,这些概念是本书所讨论的基础。

本书是一本操作系统的入门教材,适用于本科三、四年级或研究生一年级学生。我们也希望它适用于相关工程技术人员。本书清晰地描述了操作系统的基本概念。我们假设读者熟悉数据结构基础、计算机组成和一种高级语言(如C)。本书第一章介绍了学习操作系统所需的硬件知识。本书中的代码主要采用C语言形式描述,也有一部分是Java语言形式的。不过,即使读者没有这些语言的全面知识,也能理解这些算法。

本书直观地描述了概念,也阐述了重要理论结论,但省略了形式化的证明。参考文献中给出了有关研究论文,其中有的论文首次提出并证明了这些结论,有的资料是可供进一步阅读的参考材料。本书还通过使用图表和实例来代替证明,以说明结论的正确性。

本书所描述的基本概念和算法通常是基于既有商用操作系统的,我们的目的是根据通用操作系统而不是根据特定操作系统来描述这些概念和算法。本书还提供了大量与最通用的和最新的操作系统相关的例子,如Sun Microsystems的Solaris 2; Linux; Mach; 微软公司的MS-DOS、Windows NT、Windows 2000和Windows XP; DEC VMS和TOPS-20; IBM OS/2以及Apple Mac OS X操作系统。

在本书中,当以Windows XP作为示例操作系统时,表示同时适用于Windows XP与Windows 2000两个系统。如果某项特性只存在于Windows XP中而Windows 2000中没有,那么我们会给予说明;反之亦然。

本书结构

本书的结构安排反映了笔者多年讲授“操作系统”课程的经验,同时参考了本书评审专家的反馈信息,也考虑了本书以前版本读者所提交的意见。另外,本书内容还体现了由IEEE计算机学会与ACM联合工作组所出版的Computing Curricula 2001对于讲授操作系统的建议。

在本书的支持网页上,我们提供了采用本书以不同方法讲授入门操作系统

与高级操作系统课程的若干教学大纲范本。我们建议读者循序渐进地学习各章，因为这种策略能够更全面地学习操作系统。不过，通过使用教学大纲范本，读者可以按不同的顺序学习各章（或各章中的部分内容）。

本书内容

本书主要分为八个部分。

- **概述：**第一章与第二章解释了操作系统是什么、能做什么以及它们是如何设计与构造的。这一部分讨论了操作系统的公共特性是什么，操作系统能够为用户做什么，操作系统能为计算机系统操作员做什么。这些描述主要是激励性的和解释性的内容。在这些章节中，避免讨论这些问题的内部细节。因此，这部分适合于那些需要学习操作系统是什么而无需了解其内部算法细节的低年级学生或有关人员。
- **进程管理：**第三章到第七章描述了作为现代操作系统核心的进程以及并发的概念。进程是系统的工作单元。一个系统由一些并发执行的进程组成，其中一部分是操作系统进程（执行系统代码），另一部分是用户进程（执行用户代码）。这一部分包括进程调度方法、进程间通信、进程同步及死锁处理等内容。另外还讨论了有关线程的知识。
- **内存管理：**第八章和第九章主要讨论进程执行过程中的主存管理问题。为了改善 CPU 的使用率及其对用户的响应速度，计算机必须将多个进程同时保存在内存中。内存管理方案有很多，反映了各种不同的方法，某种特定算法的有效性与应用情形有关。
- **存储管理：**第十章到第十三章描述了现代计算机系统如何处理文件系统、大容量存储器与 I/O 设备。文件系统为在线存储和访问驻留在磁盘上的数据和程序提供了必要的机制。这些章节介绍了存储管理内部所使用的经典算法和结构，并对这些算法进行了深入的讨论，包括其特性、优点和缺点。由于与计算机相连的 I/O 设备的多样性，操作系统需要为应用程序提供足够的功能来控制这些设备。这部分深入讨论了系统 I/O，包括 I/O 系统设计、接口及系统内部的结构和功能。在许多方面，I/O 设备都是计算机系统中最慢的部分。由于 I/O 设备是性能瓶颈，所以也讨论了性能问题。另外，还讨论了与二级和三级存储有关的问题。
- **保护和安全：**第十四章和第十五章讨论了为使操作系统中的进程活动彼此之间不会互相影响，如何对系统中的进程加以保护。出于保护和安全的目的，我们采用了这样一种机制：确保只有获得操作系统授权的进程才可以操纵相应的文件、内存、CPU 和其他资源。保护是一种用来控制访问程序、进程或用户访问计算机系统资源的机制。这种机制必须提供声明控制和实施控制的方法。安全是指保护系统所存储的

信息（数据和代码）和计算机系统的其他物理资源以避免未经授权的访问、恶意破坏或修改以及意外出现的不一致。

- **分布式系统：**第十六章到第十八章讨论了一组不共享内存或时钟的处理器——分布式系统。分布式系统允许用户访问系统所维护的各种资源，对共享资源的访问能提高计算速度、改善数据的可用性和可靠性。这类系统也能为用户提供分布式文件系统，这是一种文件服务系统，且其用户、服务器和存储设备位于分布式系统的各处。分布式系统必须能提供各种机制以实现进程同步和通信，并处理死锁问题和集中式系统所未曾遇到的各种故障。
- **专用系统：**第十九章和第二十章讨论了专用系统，包括实时系统与多媒体系统。这些系统与本书主要讨论的通用系统不同，存在特殊需求。实时系统不但要求计算结果“正确”，还需要在规定时间内产生结果。多媒体系统要求确保服务质量，以保证多媒体数据在规定时间内提交给用户。
- **案例研究：**本书的第二十一章到第二十三章和网络上的附录 A 到附录 C 通过描述实际操作系统，融合了本书所讲解的概念。这些系统包括 Linux、Windows XP、FreeBSD、Mach 和 Windows 2000。我们选择 Linux 和 FreeBSD 是因为 UNIX 虽小但足以用于理解操作系统的内涵，而且不是“玩具”操作系统。其多数内部算法的选择主要是基于其简单的特性，而非速度或复杂度。在计算机科学系，通常可以很容易得到 Linux 和 FreeBSD 系统，因此许多学生都可以接触到这些系统。我们选择 Windows XP 和 Windows 2000 是因为它们提供了一个研究现代操作系统的机会，其设计和实现与 UNIX 操作系统有很大差异。第二十三章还简要地描述了其他一些有影响的操作系统。

操作系统环境

本书采用了许多实际操作系统的例子来说明操作系统的基本概念。其中，重点放在微软公司的操作系统产品系列（包括 Windows NT、Windows 2000 及 Windows XP）和各种版本的 UNIX（包括 Solaris、BSD 及 Mac OS X）。另外，还介绍了反映当前最新内核的 Linux V2.6 操作系统。

本书提供了许多以 C 语言和 Java 语言编写的例子。这些程序可在以下编程环境中运行。

- **Windows 系统：**基于 Windows 系统的主要编程环境是 Win32 API（Application Programming Interface，应用程序接口），它包括管理进程、线程、内存和外部设备的完整函数集。本书提供了多个 C 语言程序以说明 Win32 API 的使用方法。这些程序示例都已在 Windows 2000 和 Windows XP 系统中测试通过。

- **POSIX:** POSIX (Portable Operating System Interface, 可移植操作系统接口) 代表一组基于 UNIX 操作系统的标准接口。虽然 Windows XP 和 Windows 2000 操作系统也能运行某些 POSIX 程序, 但这里有关 POSIX 的讨论主要基于 UNIX 和 Linux 系统。POSIX 兼容系统必须实现 POSIX 核心标准 (POSIX.1), 如 Linux、Solaris 和 Mac OS X 等。POSIX 还定义了一些扩展, 如实时扩展 (POSIX.1.b)、线程库扩展 (POSIX.1.c, 也称为 Pthread)。本书提供了一些 C 语言编程实例以说明基本 POSIX API、线程 API 及实时编程 API。这些程序示例均采用 gcc 3.3 编译器在 Debian Linux 2.4 与 2.6 系统、Mac OS X 和 Solaris 9 上测试通过。
- **Java:** Java 是一种使用广泛的编程语言, 它具有丰富的 API, 并有针对线程创建和管理的内置语言支持。Java 程序可运行于任何操作系统, 只要其支持 Java 虚拟机 (Java Virtual Machine, JVM) 即可。本书提供了许多 Java 程序来说明各种操作系统概念和网络概念, 这些程序均在 Java 1.4 JVM 上测试过。

我们选择这三种编程环境, 因为我们认为它们代表了两大主流操作系统: Windows 与 UNIX/Linux 以及广为使用的 Java 环境。大多数编程示例都是用 C 写的, 希望读者能适应这种语言; 对 C 和 Java 语言熟悉的读者应很容易理解本书中的例子。

在有些情况 (如线程创建) 下, 我们采用了三种编程环境来说明某个概念, 以便使读者能在实现同一任务时对比三种不同的做法。在另外一些情况下, 可能只采用一种 API 来说明某个概念。例如, 共享内存只通过 POSIX API 来加以说明; TCP/IP 套接字编程是采用 Java API 来展示的。

第七版

在写第七版时, 不但采纳了读者对以前版本的许多评述和建议, 而且还加入一些现代操作系统和网络发展的新概念。笔者对绝大多数章节中的内容进行了改写以反映最新变化, 删除了不再适用或无关的内容。

笔者对许多章节都做了大量改写和重新组织。最为重要的是, 我们重新组织了第一章和第二章的概述部分, 并增加了两章专用系统 (实时嵌入式系统与多媒体系统) 的内容。由于保护与安全在操作系统中已越来越普及, 现在将这些内容提前了。而且, 我们对安全部分的内容做了大量修改和扩充。

下面简要介绍对各章所做的主要修改。

- **第一章, 导论,** 已完全修订。以前版本的第一章描述了操作系统发展的历史。本版的第一章概述了操作系统的主要组成部分, 并简要介绍了计算机系统的组成。
- **第二章, 操作系统结构,** 是前一版第三章的修订, 主要加强了对系统

调用和操作系统结构的讨论，也更新了有关虚拟机的内容。

- **第三章，进程**，是原来的第四章。本章增加了有关进程在 Linux 中是如何表示的内容，并采用 POSIX 和 Win32 API 来说明进程创建的方法。通过一个 POSIX 系统共享内存 API 的例程，加强了对共享内存的说明。
- **第四章，线程**，是原来的第五章。本章强化了对线程库的讨论，包括 POSIX、Win32 API 和 Java 线程库。另外，还更新了对 Linux 的多线程的描述。
- **第五章，CPU 调度**，是原来的第六章。本章更新了许多有关多处理器系统调度问题的讨论，包括处理器亲合力与负荷平衡算法。另外新增加了线程调度，包括 Pthread 和 Solaris 中的表格驱动调度。同时也修改了 Linux 调度的内容，以反映 2.6 版内核中所使用的调度。
- **第六章，进程同步**，是原来的第七章。本章删除了双进程解决方案的内容，现在只讨论 Peterson 解决方案，这是因为双进程算法并不保证能在现代处理器上正确执行。另外本章还增加了 Linux 内核和 Pthread API 中的同步的内容。
- **第七章，死锁**，是原来的第八章。本章增加了一个程序例子，用以说明多线程 Pthread 程序中的死锁处理。
- **第八章，主存**，是原来的第九章。本章不再讨论覆盖。另外，对于分段部分也做了很大修改，包括加强对 Pentium 系统中的分段的讨论，此外还描述了如何为这样的分段系统设计 Linux。
- **第九章，虚拟内存**，是原来的第十章。本章扩充了内存映射文件以引入虚拟内存，还增加了一个基于 Win32 API 的编程实例来说明通过内存映射文件共享内存的方法。更新了对于内存管理硬件细节的讨论。另外还增加了有关内核内存分配的内容，并描述了伙伴算法与分片算法。
- **第十章，文件系统接口**，是原来的第十一章。除了内容更新之外，本章还增加了一个 Windows XP ACL 的例子。
- **第十一章，文件系统实现**，是原来的第十二章。新增了对 WAFL 文件系统和 Sun ZFS 文件系统的全面描述。
- **第十二章，大容量存储结构**，是原来的第十四章。新增了现代存储阵列的内容，包括新 RAID 技术与特性如精简自动配置。
- **第十三章，I/O 系统**，是原来的第十三章，增加了一些新内容。
- **第十四章，保护**，是原来的第十八章，更新了有关最少权限原则的内容。
- **第十五章，安全**，是原来的第十九章。本章各节做了大量修改。举例说明了如何利用缓存溢出，并扩充了威胁、加密与安全工具的内容。
- **第十六章到第十八章**分别是原来的第十五章到第十七章，并扩充了一些新内容。

- **第十九章，实时系统**，是新增章，着重描述实时与嵌入式计算系统，这些系统与许多传统系统相比具有不同的需求。本章概述了实时计算机系统，并描述如何构造操作系统以满足这些系统的严格定时要求。
- **第二十章，多媒体系统**，是新增章，详细描述了多媒体系统这一新兴领域的发展。多媒体数据与传统数据不同，因为多媒体数据，如一帧图像，必须按照一定的时间限制（流式）传输。本章讨论这些要求对操作系统设计方面的影响。
- **第二十一章，Linux 系统**，是原来的第二十章，并经更新以反映最新的 Linux 2.6 内核（本书写作时的最新版本）中的变化。
- **第二十二章，Windows XP** 也做了更新。
- **第二十三章，著名操作系统** 也做了更新。

原第二十一章（Windows 2000）已改为附录 C。与前一版一样，这些附录也可在线获取。

编程练习与项目

为了巩固本书介绍的概念，我们增加了一些编程练习和项目，可利用 POSIX、Win32 API 和 Java 来完成。本书新增了 15 个编程练习，以强调进程、线程、共享内存、进程同步和网络。另外，本书增加了比一般编程练习更为复杂的几个编程项目。这些项目包括：向 Linux 内核添加系统调用，采用 `fork()` 系统调用编写 UNIX 命令解释器，多线程矩阵应用程序以及采用共享内存的生产者—消费者问题。

教学辅助材料和网页

本书的网页包含许多材料，如本书的一套幻灯片、多种教学大纲范本、C 与 Java 的源代码以及最新勘误表。网页还包括三个案例研究附录及分布式通信的附录。本网站 URL 为

<http://www.os-book.com>

本书新增了称为“学生答案手册”的书面补充材料，其中包括问题与习题的解答（这些内容不包含在本书中），以帮助学生掌握所介绍的概念。读者可以通过登录 <http://www.wiley.com/college/silberschatz> 选择“学生答案手册”链接来购买本手册。

为了获得受限资源，如本书习题的解答指南，请与当地 John Wiley & Sons 公司销售代理联系。注意：这些受限资源只有使用本书的教师才可获得。读者可通过“Find a Rep?”网页（<http://www.jsw-edcv.wiley.com/college/findarep>）来找到你的代理。

邮件列表

为了方便《操作系统概念》一书读者之间的交流，我们切换至邮件系统。如果你希望使用这一工具，请访问如下 URL 并按提示订阅：

<http://mailman.cs.yale.edu/mailman/listinfo/os-book-list>

本邮件列表系统提供了诸多便利，如记录存档，还有一些订阅选项如只订文摘或网页。如需向该列表发送消息，请将电子邮件发送至：

os-book-list@cs.yale.edu

根据邮件内容，我们可能会对某人进行回复，也可能将信息转至邮件列表上的所有用户。本列表是受控的，所以，你不会收到不适当的邮件。

使用本书作为教材上课的学生请不要通过本邮件列表来询问习题答案。对此不会予以提供。

建议

我们已设法消除新版中的所有错误，但是与操作系统一样，一些隐蔽的错误可能仍然存在。望来函告知你所发现的任何关于本书的文字错误或疏漏，我们将非常感谢。

如果你希望提供改进建议或习题，那么我们也非常期待您的来信。请发送邮件至 os-book@cs.yale.edu。

致谢

本书是根据先前版本修订而来的，前三版是与 James Peterson 一起合著的。以前版本的其他帮助者包括：Hamid Arabnia, Rida Bazzi, Randy Bentson, David Black, Joseph Boykin, Jeff Brumfield, Gael Buckley, Roy Campbell, P. C. Capon, John Carpenter, Gil Carrick, Thomas Casavant, Ajoy Kumar Datta, Joe Deck, Sudarshan K. Dhall, Thomas Doeppner, Caleb Drake, M. Racsit Eskicioğlu, Hans Flack, Robert Fowler, G. Scott Graham, Richard Guy, Max Hailperin, Rebecca Hartman, Wayne Hathaway, Christopher Haynes, Bruce Hillyer, Mark Holliday, Ahmed Kamel, Richard Kieburtz, Carol Kroll, Morty Kwestel, Thomas LeBlanc, John Leggett, Jerrold Leichter, Ted Leung, Gary Lippman, Carolyn Miller, Michael Molloy, Yoichi Muraoka, Jim M. Ng, Banu Özden, Ed Posnak, Boris Putanec, Charles Qualline, John Quarterman, Mike Reiter, Gustavo Rodriguez-Rivera, Carolyn J. C. Schauble, Thomas P. Skinner, Yannis Smaragdakis, Jesse St. Laurent, John Stankovic, Adam Stauffer, Steven Stepanek, Hal Stern, Louis Stevens, Pete Thomas, David Umbaugh, Steve Vinoski, Tommy Wagner, Larry L. Wear, John

Werth, James M. Westall, J. S. Weston 和 Yang Xiang。

第十二章的部分内容取自 Hillyer 和 Silberschatz 的一篇论文[1996]。第十七章的部分内容取自 Levy 和 Silberschatz 的一篇论文[1990]。第二十一章取自 Stephen Tweedie 未发表的一篇手稿。第二十二章取自 Dave Probert、Cliff Martin 与 Avi Silberschatz 合作的一篇未发表的手稿。附录 C 取自 Cliff Martin 未发表的一篇手稿。Cliff Martin 还帮助更新了 UNIX 附录，以描述 FreeBSD。Mike Shapiro、Bryan Cantrill 与 Jim Mauro 回答了多个 Solaris 相关问题。Josh Dees 与 Rob Reynolds 介绍了微软公司的.NET。设计与增强 UNIX 外壳接口项目由 John Trono（美国佛蒙特州 Winooski 的 St. Michael 学院）提供。

本版新增了许多习题及相应解答，这是由 Arvind Krishnamurthy 提供的。

我们感谢审阅了本版的以下各位：Bart Childs, Don Heller, Dean Hougen Michael Huang, Morty Kewstel, Euripides Montagne 和 John Sterling。

顾问编辑 Bill Zobrist 和 Paul Crockett 在准备本书期间给予了专家级指导。他们都得到了 Simon Durkin 的帮助，Simon Durkin 具体负责本项目顺利实施的诸多细节。高级制作编辑是 Ken Santor。封面绘制者是 Susan Cyr，封面设计者是 Madelyn Lesure。Beverly Peavler 审阅了手稿。自由职业校对是 Katrina Avery，自由职业索引是 Rosemary Simpson。Marilyn Turnamian 协助制作了插图和幻灯片。

最后，我们还希望感谢一些人。Avi 回到了学术界并与 Valerie 一起开始了新生活。这使他专注于写作。Pete 要感谢家人、朋友及同事在项目期间的支持和理解。Greg 要感谢家人一直以来的支持和关心。不过，他要特别感谢好友 Peter Ormsby，无论生活有多么忙碌，Peter Ormsby 总是要问“写作进展如何？”

Abraham Silberschatz, New Haven, CT, 2004

Peter Baer Galvin, Burlington, MA, 2004

Greg Gagne, Salt Lake City, UT, 2004

Preface

Operating systems are an essential part of any computer system. Similarly, a course on operating systems is an essential part of any computer-science education. This field is undergoing rapid change, as computers are now prevalent in virtually every application, from games for children through the most sophisticated planning tools for governments and multinational firms. Yet the fundamental concepts remain fairly clear, and it is on these that we base this book.

We wrote this book as a text for an introductory course in operating systems at the junior or senior undergraduate level or at the first-year graduate level. We hope that practitioners will also find it useful. It provides a clear description of the *concepts* that underlie operating systems. As prerequisites, we assume that the reader is familiar with basic data structures, computer organization, and a high-level language, such as C. The hardware topics required for an understanding of operating systems are included in Chapter 1. For code examples, we use predominantly C, with some Java, but the reader can still understand the algorithms without a thorough knowledge of these languages.

Concepts are presented using intuitive descriptions. Important theoretical results are covered, but formal proofs are omitted. The bibliographical notes contain pointers to research papers in which results were first presented and proved, as well as references to material for further reading. In place of proofs, figures and examples are used to suggest why we should expect the result in question to be true.

The fundamental concepts and algorithms covered in the book are often based on those used in existing commercial operating systems. Our aim is to present these concepts and algorithms in a general setting that is not tied to one particular operating system. We present a large number of examples that pertain to the most popular and the most innovative operating systems, including Sun Microsystems' Solaris; Linux; Mach; Microsoft MS-DOS, Windows NT, Windows 2000, and Windows XP; DEC VMS and TOPS-20; IBM OS/2; and Apple Mac OS X.

In this text, when we refer to Windows XP as an example operating system, we are implying both Windows XP and Windows 2000. If a feature exists in Windows XP that is not available in Windows 2000, we will state this explicitly.

If a feature exists in Windows 2000 but not in Windows XP, then we will refer specifically to Windows 2000.

Organization of This Book

The organization of this text reflects our many years of teaching operating systems courses. Consideration was also given to the feedback provided by the reviewers of the text, as well as comments submitted by readers of earlier editions. In addition, the content of the text corresponds to the suggestions from *Computing Curricula 2001* for teaching operating systems, published by the Joint Task Force of the IEEE Computing Society and the Association for Computing Machinery (ACM).

On the supporting web page for this text, we provide several sample syllabi that suggest various approaches for using the text in both introductory and advanced operating systems courses. As a general rule, we encourage readers to progress sequentially through the chapters, as this strategy provides the most thorough study of operating systems. However, by using the sample syllabi, a reader can select a different ordering of chapters (or subsections of chapters).

Content of This Book

The text is organized in eight major parts:

- **Overview.** Chapters 1 and 2 explain what operating systems *are*, what they *do*, and how they are *designed* and *constructed*. They discuss what the common features of an operating system are, what an operating system does for the user, and what it does for the computer-system operator. The presentation is motivational and explanatory in nature. We have avoided a discussion of how things are done internally in these chapters. Therefore, they are suitable for individual readers or for students in lower-level classes who want to learn what an operating system is without getting into the details of the internal algorithms.
- **Process management.** Chapters 3 through 7 describe the process concept and concurrency as the heart of modern operating systems. A *process* is the unit of work in a system. Such a system consists of a collection of *concurrently* executing processes, some of which are operating-system processes (those that execute system code) and the rest of which are user processes (those that execute user code). These chapters cover methods for process scheduling, interprocess communication, process synchronization, and deadlock handling. Also included under this topic is a discussion of threads.
- **Memory management.** Chapters 8 and 9 deal with main memory management during the execution of a process. To improve both the utilization of the CPU and the speed of its response to its users, the computer must keep several processes in memory. There are many different memory-management schemes, reflecting various approaches to memory management, and the effectiveness of a particular algorithm depends on the situation.

- **Storage management.** Chapters 10 through 13 describe how the file system, mass storage, and I/O are handled in a modern computer system. The file system provides the mechanism for on-line storage of and access to both data and programs residing on the disks. These chapters describe the classic internal algorithms and structures of storage management. They provide a firm practical understanding of the algorithms used—the properties, advantages, and disadvantages. Since the I/O devices that attach to a computer vary widely, the operating system needs to provide a wide range of functionality to applications to allow them to control all aspects of the devices. We discuss system I/O in depth, including I/O system design, interfaces, and internal system structures and functions. In many ways, I/O devices are also the slowest major components of the computer. Because they are a performance bottleneck, performance issues are examined. Matters related to secondary and tertiary storage are explained as well.
- **Protection and security.** Chapters 14 and 15 discuss the processes in an operating system that must be protected from one another's activities. For the purposes of protection and security, we use mechanisms that ensure that only processes that have gained proper authorization from the operating system can operate on the files, memory, CPU, and other resources. Protection is a mechanism for controlling the access of programs, processes, or users to the resources defined by a computer system. This mechanism must provide a means of specifying the controls to be imposed, as well as a means of enforcement. Security protects the information stored in the system (both data and code), as well as the physical resources of the computer system, from unauthorized access, malicious destruction or alteration, and accidental introduction of inconsistency.
- **Distributed systems.** Chapters 16 through 18 deal with a collection of processors that do not share memory or a clock—a *distributed system*. By providing the user with access to the various resources that it maintains, a distributed system can improve computation speed and data availability and reliability. Such a system also provides the user with a distributed file system, which is a file-service system whose users, servers, and storage devices are dispersed among the sites of a distributed system. A distributed system must provide various mechanisms for process synchronization and communication and for dealing with the deadlock problem and a variety of failures that are not encountered in a centralized system.
- **Special-purpose systems.** Chapters 19 and 20 deal with systems used for specific purposes, including real-time systems and multimedia systems. These systems have specific requirements that differ from those of the general-purpose systems that are the focus of the remainder of the text. Real-time systems may require not only that computed results be “correct” but also that the results be produced within a specified deadline period. Multimedia systems require quality-of-service guarantees ensuring that the multimedia data are delivered to clients within a specific time frame.
- **Case studies.** Chapters 21 through 23 in the book, and Appendices A through C on the website, integrate the concepts described in this book by describing real operating systems. These systems include Linux, Windows