

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

——Java实现

(第七版 影印版)

OPERATING SYSTEM CONCEPTS
WITH JAVA
(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

操作系统概念

——Java 实现

(第七版 影印版)

OPERATING SYSTEM CONCEPTS

WITH JAVA

(Seventh Edition)

江苏工业学院图书馆

■ Abraham Silberschatz
Peter Baer Galvin

藏书章

Greg Gagne



高等教育出版社
Higher Education Press

图字：01-2006-7473 号

Operating System Concepts with Java , 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 © 2007 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:Java 实现 =Operating System Concepts with Java: 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, 2007.4

ISBN 978-7-04-021509-0

I. 操… II. ①西… ②高… ③加… III. ①操作系统—高等学校—教材—英文②Java 语言—程序设计—高等学校—教材—英文 IV. TP316 TP312

中国版本图书馆 CIP 数据核字(2007)第 036270 号

出版发行	高等教育出版社	购书热线	010-58581118
社 址	北京市西城区德外大街 4 号	免费咨询	800-810-0598
邮政编码	100011	网 址	http://www.hep.edu.cn
总 机	010-58581000		http://www.hep.com.cn
		网上订购	http://www.landaco.com
经 销	蓝色畅想图书发行有限公司		http://www.landaco.com.cn
印 刷	北京民族印刷厂	畅想教育	http://www.widedu.com
开 本	787 × 1092 1/16	版 次	2007 年 4 月第 1 版
印 张	62.5	印 次	2007 年 4 月第 1 次印刷
字 数	1 200 000	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 21509-00

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等教育出版社的共同组织下,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、内容先进而备受欢迎。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 the two new women in my life,
Valerie and Nicolette*

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、属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展着。不过，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。我们也希望它适合相关工程技术人员。本书清晰地描述了操作系统的基本概念。作为前提，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，最好是 Java。本书第 1 章包括了学习操作系统所需的硬件知识。对于示例代码，我们主要采用了 Java，有时也使用 C。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例子来代替证明，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的商用操作系统，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, 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 和 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，那么我们会直接说明；反之亦然。

本书结构

本书的结构是根据笔者多年“操作系统”课程的讲授经验来安排的，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，读者则可以按不同地顺序学习各章（或各章中的部分内容）。

本书内容

本书共有八大部分:

- **概述。**第1章与第2章解释操作系统是什么、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不需要知道其内部算法细节的低年级学生或有关人员。
- **进程管理。**第3章到第7章描述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,其中一些是操作系统进程(执行系统代码),另一些是用户进程(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8章和第9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。内存管理方法有很多,某种特定算法的有效性与应用情形有关。
- **存储管理。**第10章到第13章描述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。
- **保护和安全的。**第14章和第15章讨论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。这种机制必须提供表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 16 章到第 18 章讨论一组不共享内存或时钟的处理器——分布式系统。这类系统允许用户访问系统所维护的各种资源。分布式系统能加快计算速度，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个场所。分布式系统必须为进程同步和通信提供各种机制，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 19 章和第 20 章讨论专用系统，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，有特殊需求。实时系统不但需要计算结果正确，还需要在规定时限内产生结果。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帧内提供给用户。
- **案例研究。**本书的第 21 章到第 23 章和网站上的附录 A 到附录 C 通过描述实际操作系统，融合了本书的概念。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不是速度或复杂度。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 操作系统的有很大不同。第 23 章简要地介绍其他一些有影响的操作系统。网站上还包括附录 D，它较本书更为详细地介绍了分布式通信的内容。

采用 Java 实现的第七版

本书采用 Java 语言来阐述许多操作系统的概念，如多线程、CPU 调度、进程同步、死锁、内存及文件管理、安全和分布式系统等。与其说是一种程序设计语言，Java 更是一种技术，因此，它是一种很优秀的论证工具。

本书所采用的许多与 Java 相关的材料均在本科生的“操作系统”课程中得到改进和课堂测试。根据笔者的经验，具有 C++ 和面向对象理论基础知识的学生，即便没有 Java 方面的知识，在学习本书时一般对 Java 也没有太大的问题。相反地，问题主要存在于对多线程、多重数据共享、并发运行线程等概念的理解上。这些概念是系统中的，而不是专门针对 Java 的；即便具有丰富 Java 知识的学生也可能存在这些困难。因此，我们强调操作系统的概念，而非专注于 Java 语法。

Java 是一项发展很快的技术；但是，我们所使用的 Java 例子对语言而言是很基本的，而且近期不大可能改变。本书的所有 Java 程序遵循 Java SDK 1.5（或 Java 5）版本。Java 5 是写此书时的最新版本，具有一些新的特点——无论是在语言级，还

是 API 级——加强了研究操作系统的语言有效性。贯穿本书，我们对 Java 5 API 增加了一些新的内容；但是，我们并没有选择 Java 新的语言特征（如 Generics、Autoboxing、增强型“for”循环），因为顾及现实因素，许多学生可能不熟悉这些新的语法。尽管你不需要用全面的 Java 知识来理解本书的例子，但如果你不了解 Java，应该读一下《Java Primer》，可以从后面的内容中找到此书的网站信息。

当我们写此《操作系统概念——Java 实现》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。

笔者对许多章节都作了大量改写和重新组织。最为重要的是，我们重新组织了第 1 章与第 2 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，所以将它们提前了。而且，我们对安全部分的内容作了大量修改与扩充。

下面简要介绍对《操作系统概念——Java 实现》第六版各章的主要修改：

- **第 1 章，导论**，已完全修改。以前版本的第 1 章描述了操作系统发展的历史。新的第 1 章概述了操作系统的主要组成部分，并简要介绍计算机系统的组成。
- **第 2 章，操作系统结构**，是前版的第 3 章的修订，主要加强了对系统调用与操作系统结构的讨论，也大大更新了虚拟机系统和 Java 操作系统部分的内容。
- **第 3 章，进程**，为原第 4 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 与 Win32 API 及 Java 来说明进程的创建。
- **第 4 章，线程**，为原第 5 章。本章强化了对线程库的讨论，包括 POSIX、Win32 API 与 Java 线程库。另外，新增 Java API 线程池的内容，并更新了对 Linux 的多线程的描述。
- **第 5 章，CPU 调度**，为原第 6 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。另增加了线程调度，包括 Pthreads 与 Solaris 的基于表格的调度。同时也修改了 Linux 调度的内容，以反映 2.6 版内核的调度。此外，还新增了 Solaris 系统中 Java 线程调度的内容。
- **第 6 章，进程同步**，为原第 7 章。本章删除了双进程解决方案的内容，仅讨论 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。此外，还增加 Linux 内核同步的内容，该章包括目前 Java 5 API 中的信号量、条件变量、Reentrant 锁的内容。
- **第 7 章，死锁**，为原第 8 章。本章增加了一个程序例子，以说明在多线程 Java 程序中采用 Reentrant 锁的死锁。
- **第 8 章，主存**，为原第 9 章。本章不再讨论覆盖。另外，对于分段部分也作了很大修改，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 9 章, 虚拟内存**, 为原第 10 章。本章涵盖了 Java 中的内存映射文件以引入虚拟内存。更新了内存管理硬件的具体内容。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 10 章, 文件系统接口**, 为原第 11 章。除了内容更新外, 本章还增加了一个 Windows XP ACL 的例子。
- **第 11 章, 文件系统实现**, 为原第 12 章。新增了 WAFL 文件系统与 Sun ZFS 文件系统的讨论。
- **第 12 章, 大容量存储结构**, 为原第 14 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 13 章, I/O 系统**, 在原第 13 章的基础上, 增加了新内容。
- **第 14 章, 保护**, 为原第 18 章, 更新了有关最少权限原则的内容。
- **第 15 章, 安全**, 为原第 19 章。本章各节作了大量修改。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 16 章到第 18 章** 分别为原第 15 章到第 17 章, 并增加了新内容。
- **第 19 章, 实时系统**, 为新增章, 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 20 章, 多媒体系统**, 为新增章, 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。本章讨论了这些要求如何影响操作系统的设计。
- **第 21 章, Linux 系统**, 为原第 20 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 22 章, Windows XP**, 也作了更新。
- **第 23 章, 著名操作系统**, 也作了更新。

原第 21 章(Windows 2000)现为附录 C。与前版一样, 这些附录也可从在线网站获取。

编程练习与项目

为了巩固本书介绍的概念, 我们增加了几个新的 Java 编程练习与项目。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、死锁、内存管理、文件系统与网络。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 内核增加新的系统调用(仅有的需用 C 语言的项目), 生成一个 UNIX 外壳, 生成多线程矩阵应用程序, 使用银行家算法避免发生死锁, 设计一个基本的文件系统以及设计一个 Web 服务器。

教学辅助材料和网页

本书的网页包括许多材料，如本书的一套幻灯片、多种教学大纲范本、所有 Java 源代码以及最新勘误表。本网页还包括了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>

本版网页新增了实践练习补充，其中包括书中所没有的练习。实践练习及相应的答案在本书的网页上公开。笔者鼓励学生自己求解实践练习，并通过网页上的答案来检查他们的解答。

为了获得受限资源，如本书习题的解答，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。笔者可以通过“Find a Rep?”网页 (<http://www.jsw-edcv.wiley.com/college/findarep>) 来找到你的代理。

邮件列表

为了方便本书读者之间的交流，我们采用了邮件系统。如果你希望获此便利，请访问如下 URL 并按要求订阅：

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

本邮件列表系统提供了许多便利，如记录存档、订阅选项（如只订文摘或网页）。如要向该列表发送消息，请将电子邮件发送至：

osj-book-list@cs.yale.edu

根据邮件内容，我们可能会单个回复，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。

使用本书上课的学生，不应通过本邮件列表来询问练习的答案。它们不会被提供的。

建议

我们已设法清除本版中的所有错误，但是与操作系统一样，一些隐蔽的错误可能仍然存在。我们非常希望听到你所发现的任何关于本书的文字错误或缺失。

如果你希望提供建议或提供习题，那么我们也非常高兴收到你的来信。请发邮件至 osj-book@cs.yale.edu。

致谢

本书根据以前版本修改而来，前三版是与 James Peterson 一起合作完成的。在以前版本中帮助我们的人还有：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 Kiebertz, 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 与 Yang Xiang.

第 12 章的部分内容取自 Hillyer 和 Silberschatz 的一篇论文[1996]。第 17 章的部分内容取自 Levy 和 Silberschatz 的一篇论文[1990]。第 21 章取自 Stephen Tweedie 未发表的手稿。第 22 章取自 Dave Probert, Cliff Martin 与 Avi Silberschatz 未发表的手稿。附录 C 取自 Cliff Martin 未发表的手稿。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。第 3 章中设计与增强 UNIX 外壳接口项目是由 John Trono 提供的。第 11 章中关于文件系统项目的最初想法则源自 Marvin Solomon。

本版新增了许多练习及相应解答, 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 Hougén, Michael Huang, Morty Kwestel, Euripides Montagne 与 John Sterling。

我们的出版者 Dan Sayre 在准备本书期间给予了专家级指导。高级制作编辑是 Ken Santor。设计由 Madelyn Lesure 监督, 封面制作是 Susan Cyr, 封面设计是 Howard Grossman。Beverly Peavler 审阅了手稿。自由职业校对是 Katrina Avery; 自由职业索引是 WordCo。Mark Wogahn 对本书的排版有所帮助。Marilyn Turnamian 帮助生成了图和幻灯片。

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。这给了他安宁, 以专注于写作。Pete 感谢家人、朋友及同事在项目期间的支持与理解。Greg 感谢家人、朋友和同事的支持与关心。

Abraham Silberschatz, New Haven, CT, 2006

Peter Baer Galvin, Burlington, MA, 2006

Greg Gagne, Salt Lake City, UT, 2006

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, preferably Java. The hardware topics required for an understanding of operating systems are included in Chapter 1. For code examples, we use predominantly Java, with some C, but the reader without a thorough knowledge of these languages can still understand the algorithms.

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 state this explicitly. If a feature exists in Windows 2000 but not in Windows XP, then we 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 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 XP, FreeBSD, Mach, and Windows 2000. We chose Linux and FreeBSD because UNIX—at one time—was almost small enough to understand yet was not a “toy” operating system. Most of its internal algorithms were

selected for *simplicity*, rather than for speed or sophistication. Both Linux and FreeBSD are readily available to computer-science departments, so many students have access to these systems. We chose Windows XP and Windows 2000 because they provide an opportunity for us to study a modern operating system with a design and implementation drastically different from those of UNIX. Chapter 23 briefly describes a few other influential operating systems. On the website, we also include Appendix D, which covers distributed communication in significantly more detail than we present in this text.

The Seventh Edition with Java

This text uses Java to illustrate many operating-system concepts, such as multithreading, CPU scheduling, process synchronization, deadlock, memory and file management, security, and distributed systems. Java is more a technology than a programming language, so it is an excellent vehicle for demonstrations.

Much of the Java-related material in this text has been developed and class-tested in undergraduate operating-systems classes. From our experience, students entering these classes lacking knowledge of Java—but with experience using C++ and basic object-oriented principles—generally have little trouble with Java. Rather, most difficulties lie in understanding such concepts as multithreading and data sharing by multiple, concurrently running threads. These concepts are systematic rather than being specific to Java; even students with a sound knowledge of Java are likely to have difficulty with them. We thus emphasize concepts of operating systems rather than concentrating on Java syntax.

Java is an especially fast-moving technology; however, we use Java examples that are fundamental to the language and are unlikely to change in the near future. All the Java programs in this text compile with Release 1.5 (or Java 5) of the Java Software Development Kit (SDK). Java 5 represents the most recent release at the time of writing and provides several new features—both at the language level and at the API level—that enhance the language’s usefulness for studying operating systems. We include several new additions to the Java 5 API throughout this text; however, we have opted not to include Java’s new language features (such as generics, autoboxing, and enhanced “for” loops) for the practical reason that many students may not yet be familiar with this new syntax. Although you do not need a thorough knowledge of Java to understand the examples in the text, if you do not know Java, you should read the Java Primer, which you can find at the website for this book, described below.

As we wrote this seventh edition of *Operating System Concepts with Java*, we were guided by the many comments and suggestions of readers of our previous editions, as well as by our own observations about the rapidly changing fields of operating systems and networking. We have rewritten the material in most of the chapters by bringing older material up to date and removing material that was no longer of interest or relevance.

We have made substantive revisions and organizational changes in many of the chapters. Most importantly, we have completely reorganized the overview