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

实用操作系统概念

（影印版）

APPLIED OPERATING SYSTEM CONCEPTS

Abraham Silberschatz
Peter Galvin
Greg Gagne

高等教育出版社
Higher Education Press
John Wiley & Sons, Inc.

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

实用操作系统概念

（影印版）

APPLIED OPERATING SYSTEM CONCEPTS

Abraham Silberschatz
Peter Galvin
Greg Gagne

高 等 教 育 出 版 社
John Wiley & Sons, Inc.

图字：01-2001-1040 号

Applied Operating System Concepts
Abraham Silberschatz, Peter Galvin & Greg Gagne

图书在版编目(CIP)数据

实用操作系统概念：英文 / （美）西尔伯沙茨（Silberschatz,A.）等.
—北京：高等教育出版社，2001
ISBN 7-04-010052-5

I. 实… II. 西… III. 操作系统（软件）– 英文 IV. TP316

中国版本图书馆 CIP 数据核字（2001）第 19843 号

实用操作系统概念
Abraham Silberschatz 等

出版发行	高等教育出版社		
社　　址	北京市东城区沙滩后街 55 号	**邮政编码**	100009
电　　话	010-64054588	**传　　真**	010-64014048
网　　址	http:// www. hep. edu. cn http:// www. hep. com. cn		
经　　销	新华书店北京发行所		
印　　刷	北京外文印刷厂		
开　　本	787×1092　1/16	**版　　次**	2001 年 5 月第 1 版
印　　张	54	**印　　次**	2001 年 5 月第 1 次印刷
字　　数	1 209 000	**定　　价**	45.00 元

前　　言

20世纪末，以计算机和通信技术为代表的信息科学和技术，对世界的经济、军事、科技、教育、文化、卫生等方面的发展产生了深刻的影响，由此而兴起的信息产业已经成为世界经济发展的支柱。进入21世纪，各国为了加快本国的信息产业，加大了资金投入和政策扶持。

为了加快我国信息产业的进程，在我国《国民经济和社会发展第十个五年计划纲要》中，明确提出"以信息化带动工业化，发挥后发优势，实现社会生产力的跨越式发展。"信息产业的国际竞争将日趋激烈。在我国加入WTO后，我国信息产业将面临国外竞争对手的严峻挑战。竞争成败最终将取决于信息科学和技术人才的多少与优劣。

在20世纪末，我国信息产业虽然得到迅猛发展，但与国际先进国家相比，差距还很大。为了赶上并超过国际先进水平，我国必须加快信息技术人才的培养，特别要培养一大批具有国际竞争能力的高水平的信息技术人才，促进我国信息产业和国家信息化水平的全面提高。为此，教育部高等教育司根据教育部吕福源副部长的意见，在长期重视推动高等学校信息科学和技术的教学的基础上，将实施超前发展战略，采取一些重要举措，加快推动高等学校的信息科学和技术等相关专业的教学工作。在大力宣传、推荐我国专家编著的面向21世纪和"九五"重点的信息科学和技术课程教材的基础上，在有条件的高等学校的某些信息科学和技术课程中推动使用国外优秀教材的影印版进行英语或双语教学，以缩短我国在计算机教学上与国际先进水平的差距，同时也有助于强化我国大学生的英语水平。

为了达到上述目的，在分析一些出版社已影印相关教材，一些学校已试用影印教材进行教学的基础上，教育部高等教育司组织并委托高等教育出版社开展国外优秀信息科学和技术优秀教材及其教学辅助材料的引进研究与影印出版的试点工作。为推动用影印版教材进行教学创造条件。

本次引进的系列教材的影印出版工作，是在对我国高校的信息科学和技术专业的课程与美国高校的进行对比分析的基础上展开的；所影印出版的教材均由我国主要高

校的信息科学和技术专家组成的专家组，从国外近两年出版的大量最新教材中精心筛选评审通过的内容新、有影响的优秀教材；影印教材的定价原则上应与我国大学教材价格相当。

教育部高等教育司将此影印系列教材推荐给高等学校，希望有关教师选用，使用后有什么意见和建议请及时反馈。也希望有条件的出版社，根据影印教材的要求，积极参加此项工作，以便引进更多、更新、更好的外国教材和教学辅助材料。

同时，感谢国外有关出版公司对此项引进工作的配合，欢迎更多的国外公司关心并参与此项工作。

教育部高等教育司

二〇〇一年四月

To my mother, Wira,
and my children, Lemor, Sivan, and Aaron

Avi Silberschatz

To Carla, Gwendolyn, and Owen

Peter Galvin

To my wife, Pat,
and my sons, Tom, and Jay

Greg Gagne

PREFACE

Operating systems are an essential part of any computer system. Similarly, a course in operating systems is an essential part of any computer-science education. This book is intended to serve as a text for an introductory course in operating systems at the junior or senior undergraduate level, or at the first-year graduate level. It provides a clear description of the *concepts* that underlie operating systems.

We discuss fundamental concepts that are applicable to a variety of systems. We use Java to present many of these ideas, and we include numerous examples that pertain specifically to UNIX and to other popular operating systems. In particular, we use Sun Microsystems' Solaris 2 operating system, in addition to Microsoft Windows and Windows NT. Other operating systems discussed are Linux, IBM OS/2, the Apple Macintosh Operating System, and Mach.

Contents of This Book

The text is organized in seven major parts:

- **Overview** (Chapters 1 through 3). In these chapters, we explain what operating systems *are,* what they *do,* and how they are *designed* and *constructed.* We explain how the concept of an operating system has developed, 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, historical, and explanatory; we avoid discussing how the ideas are carried out within the computer system. Therefore, these chapters are suitable for individuals, or for students in lower-level classes, who want to learn what an operating system is without studying the details of the internal algorithms. Chapter 2 covers the hardware topics that are important to operating systems. Readers well versed in such topics (for example, I/O, DMA, and hard-disk operation) may choose to skim or skip this chapter.

- **Process management** (Chapters 4 through 8). At the heart of modern operating systems are *processes*, the units of work, which execute *concurrently*. There are operating-system processes, which execute system code, and user processes, which execute user code. These chapters cover various methods for process scheduling, interprocess communication, process synchronization, and deadlock handling. Also included under this topic is a discussion of threads.

- **Memory and storage management** (Chapters 9 through 13). A process must be (at least partially) in main memory during execution. To ensure good utilization of and response speed of the CPU, the computer must keep several processes in memory. The many different memory-management schemes reflect various approaches; their effectiveness depends on the particular situation. Since main memory is usually too small to accommodate all data and programs, and since it cannot store data permanently, the computer system must provide secondary storage to back up main memory. Most modern computer systems use disks as the primary on-line storage medium for information (both programs and data). The file system provides the mechanism for on-line storage of and access to both data and programs that reside on the disks. These chapters deal with the classic internal algorithms and structures of storage management. They provide a firm practical understanding of the properties, advantages, and disadvantages of the algorithms used. The devices that attach to a computer vary in multiple dimensions. In many ways, they are the slowest major components of the computer. Because devices vary so widely, the operating system needs to provide a wide range of functionality to allow applications to control all aspects of the devices. These chapters discuss system I/O in depth, including I/O system design, interfaces, and internal system structures and functions. Because devices create a performance bottleneck, we examine other influences on performance. Matters related to secondary and tertiary storage are explained as well.

- **Distributed systems** (Chapters 14 through 17). A *distributed system* is a collection of processors that do not share memory or a clock. Such a system gives the user access to the various resources that the system maintains. Access to a shared resource allows computation speedup and improved data availability and reliability. Such a system also gives the user a distributed file system, which is a file-service system whose users, servers, and storage devices are dispersed among the various sites of a distributed system. A distributed system must provide various mechanisms for process synchronization and communication, so that we can deal with the deadlock problem and the variety of failures that are not encountered in a centralized system.

- **Protection and security** (Chapters 18 and 19). The various processes in an operating system must be protected from one another's activities. For that purpose, we use mechanisms to ensure that the files, memory segments, CPU, and other resources can be operated on by only those processes that have gained proper authorization from the operating system. 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 for specification of 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.

- **Case studies** (Chapters 20 through 22). We present detailed case studies of three operating systems: UNIX Berkeley 4.3BSD, Linux, and Microsoft Windows NT. We chose Berkeley 4.3BSD and Linux because UNIX at one time was almost small enough to understand, and yet was not a toy operating system. Most of its internal algorithms were selected for *simplicity*, rather than for speed or sophistication. Both Berkeley 4.3BSD and Linux are readily available to computer-science departments, so many students have access to these systems. We chose Windows NT because it is a modern operating system whose design and implementation are drastically different from those of UNIX.

Java

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

Java was originally developed as a language to program microprocessors in consumer devices such as cellular telephones and set-top boxes. The advent of the Web in the mid-1990s showed the Java team at Sun Microsystems the language's potential usefulness across the Internet: Programmers can use Java to write applications and *applets*—programs that run on web pages. Java also provides support for development of database applications, graphical user interfaces (GUIs), reusable objects, and two-dimensional and three-dimensional modeling, to name a few. All such programs can run on a single computer or on a distributed system across the Internet.

We provide an overview of Java technology in Chapter 3, and illustrate the creation and coordination of multitasking Java programs in Chapters 5 and 7. In Chapter 4, we use Java to demonstrate how different processes can communicate using shared memory and message passing. In Chapter 6, we use Java to demonstrate CPU-scheduling algorithms; in Chapter 8, we illustrate deadlock and deadlock-recovery methods using Java. In Chapter 15, we show how Java can apply many of these concepts to a distributed system. To a lesser extent, we also use Java to illustrate a virtual machine as well as memory management and computer security.

Much of the Java-related material in this text has been developed and class tested in undergraduate operating-systems classes. Students typically entered these classes with knowledge of C++ and of basic object-oriented principles, but they were not familiar with Java. They had no trouble with the syntax of Java; their difficulties lay in undertsanding such concepts as multithreading and multiple concurrently running threads sharing data. These concepts are more operating-system than Java issues; even students with a sound knowledge of Java are likely to have difficulty with them. We emphasize concurrency and passing of object references to several threads, rather than concentrating on syntax.

Java is an especially fast-moving technology; however, we use Java examples that are core to the language and are unlikely to change in the near future. All the Java programs in this text compile with the Java Development Kit (JDK), Release 1.2. 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 in Appendix A.

Relationship to OSC

This text is based on the Fifth Edition of *Operating System Concepts* (OSC). Much of the material in this text was derived from OSC; the texts differ in how the material is presented. OSC presented concepts fundamental to all operating systems, rather than particular implementations of systems. This text discusses applied knowledge. We present the concepts using examples from Java and from many contemporary operating systems, such as Solaris 2 and Windows NT.

The applied nature of this text and the addition of Java material resulted in the need to remove material from OSC in memory management, process management, mass-storage systems, and distributed systems.

Mailing List and Supplements

For information about teaching supplements and on-line material, which complement this book, visit our web site at

http://www.wiley.com/college/silberschatz

To obtain restricted supplements, you should contact your local sales representative. You can find the local Wiley representative at the "Who's My Rep?" webpage: http://www.wiley.com/college/rep/index.html

We also provide an environment where users can communicate with one another and with us. Our mailing list, consisting of users of our book, has the address aos-book@research.bell-labs.com. If you want to be on the list, please send a message to avi@bell-labs.com indicating the title of the book, your name, your affiliation, and your e-mail address.

Home Page

A World Wide Web home page for the book is available at

http://www.bell-labs.com/topic/books/aos-book

The home page contains information about the book, such as the set of slides that accompanies the book, the postscript files of the Mach chapter, model course syllabi, and up-to-date errata.

Errata

We have attempted to fix every error in this new edition, but—as happens with operating systems—a few obscure bugs will remain. We would appreciate it if you would notify us of any errors or omissions that you identify in the book. Also, if you would like to suggest improvements or to contribute exercises, we would be glad to hear from you. Send your suggestions to Avi Silberschatz, Director, Information Sciences Research Center, MH 2T-310, Bell Laboratories, 600 Mountain Ave., Murray Hill, NJ 07974 (avi@bell-labs.com).

Acknowledgments

This book is derived from Operating Systems Concepts, the first three editions of which were coauthored by James Peterson. Other people who helped us with the previous editions are Randy Bentson, David Black, Joseph Boykin, Jeff Brumfield, Gael Buckley, P. C. Capon, John Carpenter, Thomas Casavant, Ajoy Kumar Datta, Joe Deck, Thomas Doeppner, Caleb Drake, M. Raşit Eskicioğlu, Hans Flack, Robert Fowler, G. Scott Graham, Rebecca Hartman, Wayne Hathaway, Christopher Haynes, Bruce Hillyer, Mark Holliday, Richard Kieburtz, Carol Kroll, Thomas LeBlanc, John Leggett, Jerrold Leichter, Ted Leung, Gary Lippman, Carolyn Miller, Michael Molloy, Yoichi Muraoka, Jim M. Ng, Ed Posnak, Boris Putanec, Charles Qualline, John Quarterman, John Stankovic, Adam Stauffer, Steven Stepanek, Hal Stern, Louis Stevens, David Umbaugh, Steve Vinoski, John Werth, and J. S. Weston. Hamid Arabnia, Sudarshan K. Dhall, Steven Stepanek, Pete Thomas, L. David Umbaugh, Tommy Wagner.

The idea of creating an applied version of the Operating System Concepts book came from Maite Suarez-Rivas. The cover illustrator was Susan Cyr while the cover designer was Lynne Reed. Michael Hirsch, executive marketing manager, has done all the promotions in preparation for the release of the text. Patricia Unubun was helpful with book production. Lyn Dupré edited the book; Barbara Pendergast did the proofreading. Marilyn Turnamian updated the text, Instructors Manual and the slides.

Michael Shapiro reviewed the Solaris information. Chapter 13 was partially derived from a paper by Hillyer and Silberschatz [1996]. Chapter 17 was derived from a paper by Levy and Silberschatz [1990]. Chapter 21 was derived from an unpublished manuscript by Stephen Tweedie. Chapter 22 was derived from an unpublished manuscript by Cliff Martin; additional editing on that chapter was done by Bruce Hillyer.

We thank the following people, who reviewed this book: Bil Lewis, Bob Erickson, Robert Gann, Chris Armen, Anthony J. Duben, Kadathur Lakshmanan, Boleslaw K. Szymanski, David Binkley, Arun Ektare, John R. Graham, Henning Schulzrinne, Robert Kline, Stephen J. Hartley, Paul W. Ross, Mike Dahlin, John Norton, James Heliotis, and Ruknet Cezzar.

Abraham Silberschatz, Murray Hill, NJ, 1999
Peter Baer Galvin, Norton MA, 1999
Greg Gagne, Salt Lake City, UT, 1999

CONTENTS

PART ONE ■ OVERVIEW

Chapter 1 Introduction

Chapter 2 Computer-System Structures

Chapter 3 Operating-System Structures

PART TWO ■ PROCESS MANAGEMENT

Chapter 4 Processes

Chapter 5 Threads

Chapter 6 CPU Scheduling

Chapter 7 Process Synchronization

Chapter 8 Deadlocks

PART THREE ■ STORAGE MANAGEMENT

Chapter 9 Memory Management

Chapter 10 Virtual Memory

Chapter 11 File Systems

Chapter 12 I/O Systems

Chapter 13 Mass-Storage Structure

PART FOUR ■ DISTRIBUTED SYSTEMS

Chapter 14 Network Structures

Chapter 15 Distributed Communication

Chapter 16 Distributed Coordination

Chapter 17 Distributed File Systems

PART FIVE ■ PROTECTION AND SECURITY

Chapter 18 Protection

Chapter 19 Security