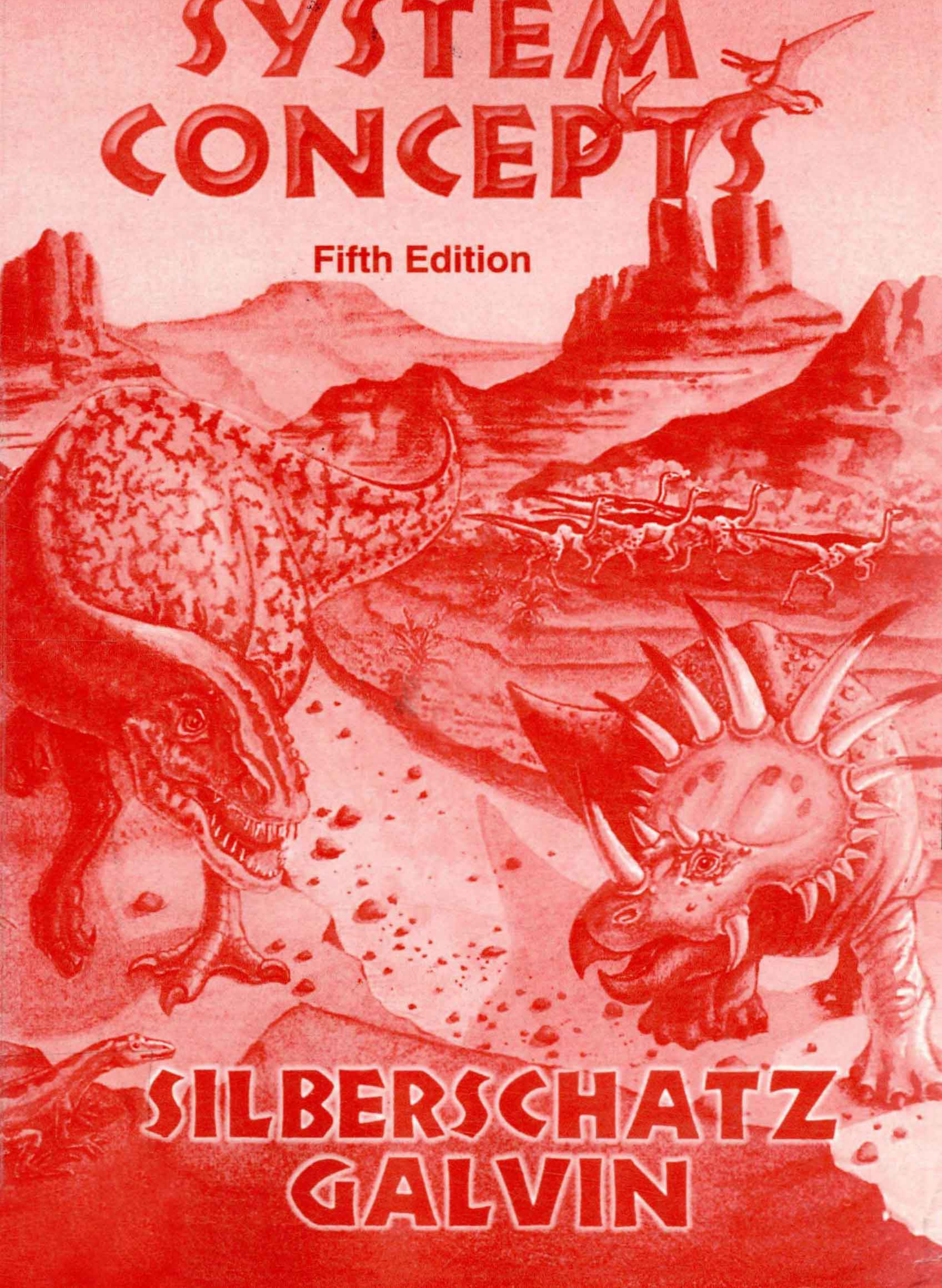
OPERATING SYSTEM CONCEPTS

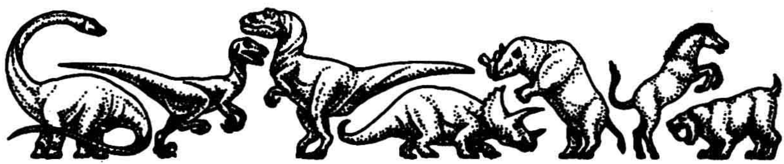
Fifth Edition



SILBERSCHATZ
GALVIN

OPERATING SYSTEM CONCEPTS

Fifth Edition



Abraham Silberschatz

Bell Labs

Peter Baer Galvin

Corporate Technologies, Inc.

John Wiley & Sons, Inc., Singapore, New York,

Chichester, Brisbane, Toronto

Library of Congress Cataloging-in-Publication Data:
Silberschatz, Abraham.
Operating system concepts / Abraham Silberschatz, Peter Galvin.—
5th ed.
p. cm.
Includes bibliographical references and index.
ISBN 9971-51-275-0
1. Operating systems (Computers) I. Galvin, Peter B. II. Title.
QA76.76.063S5583 1997
005.4, 3—dc21 97-28556
CIP

ISBN 9971-51-275-0

Printed and bound by Replika Press Pvt Ltd, 100% EOU,
Delhi-110 040 (India).

10 9 8 7 6

The scene depicted is set during the late Cretaceous period in the western United States. A *Tyrannosaurus rex* threatens a *Staurikosaurus* while a small varanid lizard is safe in a jumble of rocks. A small herd of *Ornithomimus* stays out of the way in the background while *Pterosaurs* circle in flight above. The massive *Alamosaurus* seems unconcerned by the confrontation.

To my parents, Wira and Mietek,
my wife, Haya,
and my children, Lemor, Sivan, and Aaron.

Avi Silberschatz

To Carla and Gwendolyn

Peter Galvin

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 book is intended 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.

In this book, we do not concentrate on any particular operating system or hardware. Instead, we discuss fundamental concepts that are applicable to a variety of systems. We present a large number of examples that pertain specifically to UNIX and to other popular operating systems. In particular, we use Sun Microsystem's Solaris 2 operating system, a version of UNIX, which recently has been transformed into a modern operating system with support for threads at the kernel and user levels, symmetric multiprocessing, and real-time scheduling. Other examples used include Microsoft MS-DOS, Windows, and Windows NT, Linux, IBM OS/2, the Apple Macintosh Operating System, and DEC VMS and TOPS-20.

Prerequisites

As prerequisites, we assume that the reader is familiar with general computer organization and with a high-level language, such as PASCAL. The hardware topics required for an understanding of operating systems are included in Chapter 2. We use pseudo-PASCAL notation for code examples, but the algorithms can be understood by people who do not have a thorough knowledge of PASCAL.

Content of this Book

The text is organized in seven major parts:

- **Overview** (Chapters 1 to 3). These chapters explain what operating systems *are,* what they *do,* and how they are *designed* and *constructed.* They 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 in nature. We have avoided a discussion of how things are done internally in these chapters. Therefore, they are suitable for individuals 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. Chapter 2 covers the hardware topics that are important to an understanding of operating systems. Readers well-versed in hardware topics, including I/O, DMA, and hard-disk operation, may chose to skim or skip this chapter.

- **Process management** (Chapters 4 to 7). The process concept and concurrency are at 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 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 8 to 11). A process must be in main memory (at least partially) during execution. To improve both the utilization of 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. These schemes reflect various approaches to memory management, and the effectiveness of the different algorithms 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 residing on the disks. These chapters deal with the classic internal algorithms and structures of storage management. They provide a firm practical understanding of the algorithms used—the properties, advantages, and disadvantages.

- **I/O systems** (Chapters 12 to 14). The devices that attach to a computer vary in multiple dimensions. In many ways, they are also the slowest major components of the computer. Because devices vary so widely, the operating system needs to provide a wide range of functionality to applications to allow them to control all aspects of the devices. This section discusses system I/O in depth, including I/O system design, interfaces, and internal system structures and functions. Because devices are a performance bottleneck, performance issues are examined. Matters related to secondary and tertiary storage are explained as well.

- **Distributed systems** (Chapters 15 to 18). A *distributed system* is a collection of processors that do not share memory or a clock. Such a system provides the user with 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 provides the user with 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, for dealing with the deadlock problem and the variety of failures that are not encountered in a centralized system.

- **Protection and security** (Chapters 19 and 20). The various processes in an operating system must be protected from one another's activities. For that purpose, mechanisms exist that we can use 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 21 to 24). We integrate the various concepts described in this book by describing real operating systems. Three such systems are covered in great detail—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 provides an opportunity for us to study a modern operating system that has a design and implementation drastically different from those of UNIX. Chapter 24 briefly describes a few other influential operating systems.

The Fifth Edition

Many comments and suggestions were forwarded to us concerning our previous editions. These inputs, with our own observations, have prodded us to produce this fifth edition. In particular, we have made drastic changes to our coverage of I/O systems, and have added two new chapters covering modern operating systems. Specifically, we rewrote the material in several chapters that deal with storage devices (Chapters 1, 2, 3, and 8) by bringing older material up to date and removing material that was no longer of interest. We have added new chapters on I/O systems, on tertiary storage, on Linux, and on Windows NT.

We made substantive revisions and changes in organization structure in the following chapters:

- **Chapter 3.** We have added a new section discussing the Java Virtual Machine.
- **Chapter 4.** We have added a new section describing the interprocess-communication facility of Windows NT.
- **Chapter 12.** This new chapter describes operating-system I/O architecture and implementation, including kernel structure, transfer methods, notification methods, and performance.
- **Chapter 13.** This chapter was the old Chapter 12. We have updated the material substantially.
- **Chapter 14.** This new chapter covers tertiary-storage systems.
- **Chapter 19.** This chapter was the old Chapter 13.
- **Chapter 20.** This chapter was the old Chapter 14.
- **Chapter 21.** This chapter was the old Chapter 19.
- **Chapter 22.** This new chapter covers Linux.
- **Chapter 23.** This new chapter covers Windows NT.
- **Chapter 24.** This chapter was the old Chapter 21. We have added a section describing early-batch systems (previously covered in Chapter 1), and a section summarizing the novel features of the Mach operating system.

Coverage of the Mach operating system (old Chapter 20), which is a modern operating system that provides compatibility with 4.3BSD, is available on line. Coverage of the Nachos System (old Appendix), which provides a good way to gain a deeper understanding of modern operating-systems concepts and allows students to get their hands dirty—to take apart the code for an operating system, to see how it works at a low level, to build significant pieces of the operating system themselves, and to observe the effects of their work—also is available on line.

Mailing List and Supplements

We now have a web page for this book that contains information such as the set of slides that accompanies the book, the postscript files of the Mach and Nachos chapters, and the most recent errata list. We also provide an environment where users can communicate among themselves and with us. We have created a mailing list consisting of users of our book with the following address: os-book@research.bell-labs.com. If you wish to be on the list, please send a message to avi@bell-labs.com indicating your name, affiliation, and e-mail address.

For information about teaching supplements and on-line material, which complement the book, visit the URL http://www.awl.com/cseng/books/osc5e. A list of supplements and detail contact information for the United States and international can be obtained via an automatic e-mail reply generated by sending mail to osc@aw.com. To obtain restricted supplements, please contact your local sales representative.

Errata

We have attempted to clean up every error in this new edition, but—as happens with operating systems—a few obscure bugs will remain. We would appreciate it if you, the reader, 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. Any correspondence should be sent to Avi Silberschatz, Director, Information Sciences Research Center, MH 2T-210, Bell Laboratories, 600 Mountain Ave., Murray Hill, NJ 07974 (avi@bell-labs.com).

Acknowledgments

This book is derived from the previous editions, the first three of which which were coauthored by James Peterson. Other people that have helped us with the previous editions include 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, Mark Holliday, Richard Kieburtz, Carol Kroll, Thomas LeBlanc, John Leggett, Jerrold Leichter, Ted Leung, Gary Lippman, Carolyn Miller, Michael Molloy, Yoichi Muraoka, m 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.

Lyn Dupré copyedited the book; Cliff Wilkes provided technical copyediting; Sara Strandtman edited our text into Latex format; Marilyn Turnamian helped with the generation of figures. Debbie Lafferty, Lynne Doran Cote, and Patricia Unubun were helpful with book production.

Bruce Hillyer reviewed and helped with the rewrite of Chapters 2, 12, 13, and 14. Chapter 14 was derived from a paper by Hillyer and Silberschatz [1996]. Chapter 17 was derived from a paper by Levy and Silberschatz [1990]. Chapter 19 was derived from a paper by Quarterman et al. [1985]. Chapter 22 was derived from an unpublished manuscript by Stephen Tweedie. Chapter 23 was derived from an unpublished manuscript by Cliff Martin. Additional editing on this chapter was done by Bruce Hillyer.

We thank the following people, who reviewed this edition of the book: Hamid Arabnia, Sudarshan K. Dhall, Steven Stepanek, Pete Thomas, L. David Umbaugh, Tommy Wagner.

Abraham Silberschatz, 1997, Murray Hill, NJ
Peter Baer Galvin, 1997, Norton MA

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 CPU Scheduling

Chapter 6 Process Synchronization

Chapter 7 Deadlocks

PART THREE ■ STORAGE MANAGEMENT

Chapter 8 Memory Management

Chapter 9 Virtual Memory

Chapter 10 File-System Interface

Chapter 11 File-System Implementation

PART FOUR ■ I/O SYSTEMS

Chapter 12 I/O Systems

Chapter 13 Secondary-Storage Structure

Chapter 14 Tertiary-Storage Structure

PART FIVE ■ DISTRIBUTED SYSTEMS

Chapter 15 Network Structures

Chapter 16 Distributed System Structures

Chapter 17 Distributed File Systems

Chapter 18 Distributed Coordination