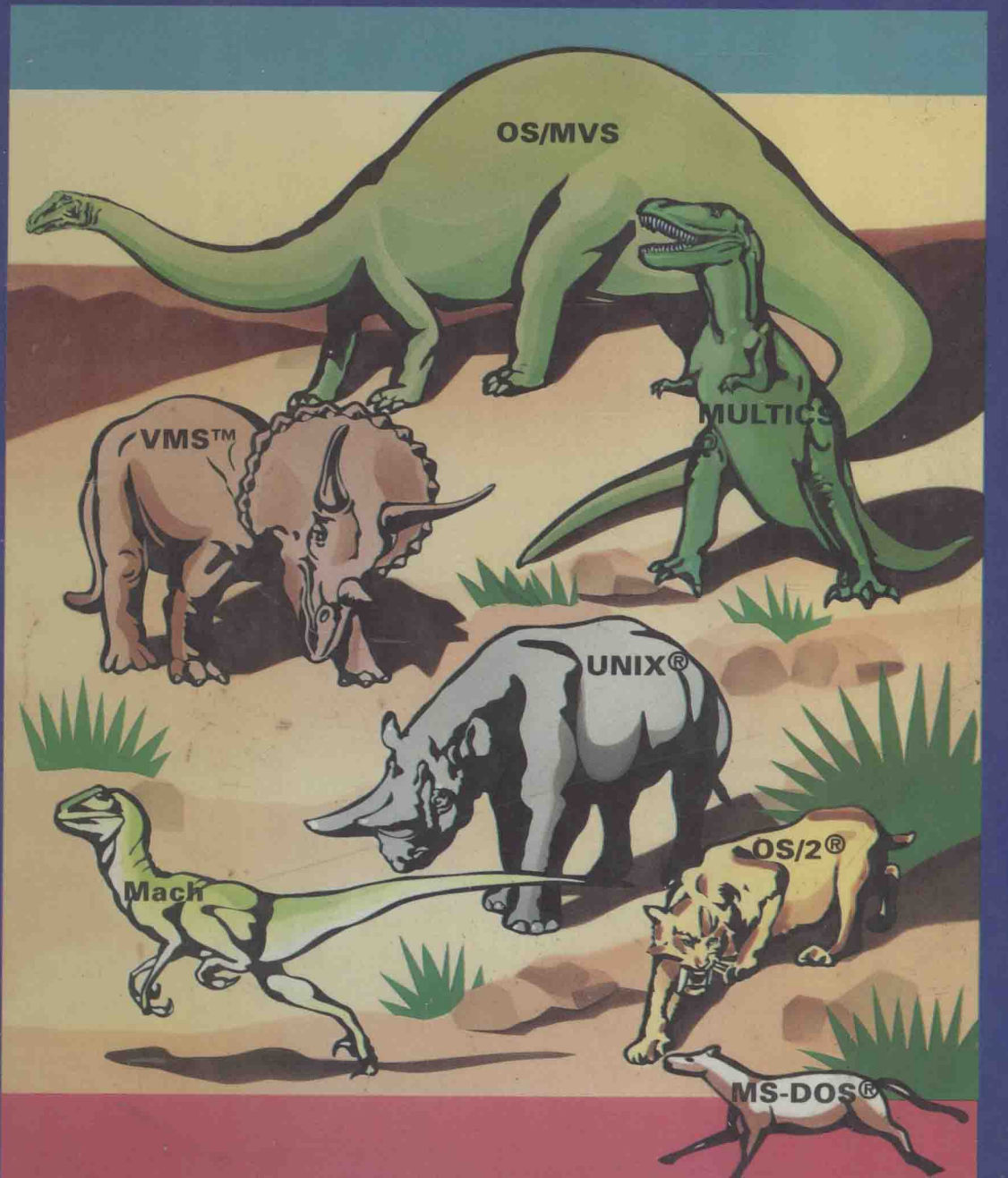
A. Silberschatz J. Peterson P. Galvin

Operating System Concepts

THIRD EDITION



Operating System Concepts THIRD EDITION

Abraham Silberschatz University of Texas at Austin

James L. Peterson IBM Corporation

Peter B. Galvin Brown University

Addison-Wesley Publishing Company

Reading, Massachusetts ▪ Menlo Park, California ▪ New York
Don Mills, Ontario ▪ Wokingham, England ▪ Amsterdam ▪ Bonn
Sydney ▪ Singapore ▪ Tokyo ▪ Madrid ▪ San Juan ▪ Milan ▪ Paris

This book is in the Addison-Wesley Series in Computer Science.
Consulting Editor: Michael Harrison

Fig. 3.8 from Iaccobucci, *OS/2 Programmer's Guide*, © 1988, McGraw-Hill, Inc., New York, New York. Fig. 1.7, p. 20. Reprinted with permission of the publisher.

Fig. 8.15 reprinted with permission from *IBM Systems Journal*, Vol. 10, No. 3, © 1971 International Business Machines Corporation.

Fig. 15.1 from Leffler/McKusick/Karels/Quarterman, *The Design and Implementation of the 4.3BSD UNIX Operating System*, © 1989, Addison-Wesley Publishing Co., Inc., Reading, Massachusetts. Fig. 1.1. Reprinted with permission of the publisher.

Figs. 16.1, 16.5, and 16.6 reproduced with permission from Open Software Foundation, Inc. Excerpted from Mach Lecture Series, OSF, October 1989, Cambridge, Massachusetts. Figs. 16.1 and 16.6 presented by R. Rashid of Carnegie-Mellon University and Fig. 16.5 presented by D. Julin of Carnegie-Mellon University.

Fig. 16.4 from Accetta/Baron/Bolosky/Golub/Rashid/Tevanian/Young, "Mach: a new kernel foundation for UNIX development," *Proceedings of Summer USENIX*, June 1986, Atlanta, Georgia. Reprinted with permission of the authors.

Library of Congress Cataloging-in-Publication Data
Silberschatz, Abraham.
Operating system concepts / Abraham Silberschatz, James L. Peterson, and Peter Galvin. -- 3rd ed.
p. cm.
Includes bibliographical references and index.
ISBN 0-201-51379-X
1. Operating systems (Computers) I. Peterson, James Lyle. II. Galvin, Peter. III. Title.
QA76.76.063S5583 1991
005.4'3--dc20 90-44561
CIP

Reprinted with corrections April, 1991

3-DO-91

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

Avi Silberschatz

To my wife, Jeanne,
and my children, Jennifer and Kathryn.

Jim Peterson

To Carla.

Peter Galvin

PREFACE

Operating systems are an essential part of a computer system. Similarly, a course on operating systems is an essential part of a 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 first-year-graduate level. It provides a clear description of the *concepts* that underlie operating systems.

This book does not concentrate on any particular operating system or hardware. Instead, it discusses fundamental concepts that are applicable to a variety of systems.

Content of this Book

As prerequisites, we assume the reader is familiar with general assembly-language programming and computer organization. The text is organized in six major parts:

- **Overview** (Chapters 1 to 3). These chapters explain what operating systems *are*, what they *do*, and how they are *designed* and *constructed*. These chapters 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 lower-level classes who want to learn

what an operating system is, without getting into the details of the internal algorithms.

- **Process management** (Chapters 4 to 6). The process concept and concurrency are at the very 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 management, CPU scheduling, process synchronization and communication, and deadlock handling.

- **Storage management** (Chapters 7 to 9). 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. The algorithms are presented in a natural order, so that new, more complicated systems can be built on the understanding of simpler systems.

- **Files and protection** (Chapters 10 and 11). A *file* is a collection of related information defined by its creator. Files are mapped, by the operating system, onto physical devices. Protection mechanisms provide controlled access by limiting the types of file access that can be made by the various users. Protection must also be available to ensure that the besides files, memory segments, CPU, and other resources can be operated on by only those processes that have gained proper authorization from the operating system.

- **Distributed systems** (Chapters 12 to 14). 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 the system maintains. Access to a shared resource allows computation speedup and improved data availability and reliability. A distributed system must provide various mechanisms for process synchronization and communication, for dealing with the deadlock problem and variety of failures that are not encountered in a centralized system.

- **Case studies** (Chapters 15 to 17). These chapters illustrate how the many concepts described can be put together in a real system. Two UNIX-based operating systems are covered in detail — Berkeley's 4.3BSD and Mach. These operating systems were chosen in part because UNIX at one time was almost small enough to understand and yet is not a toy operating system. Most of its internal

algorithms were selected for *simplicity*, not for speed or sophistication. UNIX is readily available to computer-science departments, so many students have access to it. Mach provides an opportunity for us to study a modern operating system that provides compatibility with 4.3BSD but has a drastically different design and implementation. Chapter 17 briefly describes some of the most influential operating systems.

Organization

Operating systems first began to appear in the late 1950s, and for 20 years underwent major changes in concepts and technology. As a result, the first-generation operating-system textbooks that appeared during this period (such as Brinch Hansen [1973a], Madnick and Donovan [1974], Shaw [1974], Tsichritzis and Bernstein [1974], and Habermann [1976]) tried to explain a subject that was changing even as the books were being written.

Over time, however, operating-system theory and practice appeared to mature and stabilize. The fundamental operating-system concepts seemed well defined and well understood. The basic approaches to process management, CPU scheduling, process coordination, memory management, file systems, and so on, appeared unlikely to change. At this point, a second generation of operating-systems texts appeared (such as Deitel [1983] and the first edition of this book, Peterson and Silberschatz [1983]).

For the first edition, our primary goal was to present well-understood, agreed-on, classic operating-system material. The basic concepts were organized and presented carefully; the material flowed naturally from basic principles to more sophisticated ones. The bibliographic notes contained pointers to the research papers in which results were first presented.

The fundamental concepts and algorithms covered in the first edition were often based on those used in existing commercial or experimental operating systems. Our aim was to present these concepts and algorithms in a general setting that was not tied to one particular operating system. While the general presentation in the first edition was well-received, there was a strong demand for a more complete practical example, and so in the second edition, Peterson and Silberschatz [1985] we added a chapter on the UNIX operating system.

The first and second editions were organized to stress the basic practical aspects of operating systems (CPU scheduling and memory management) before presenting the unifying theoretical concept of the process model for operating systems. An alternate edition, Silberschatz

and Peterson [1988], which reordered the presentation to discuss the process concept very early, was well-received, and eventually replaced the second edition.

The Third Edition

Many comments and suggestions were forwarded to us concerning our previous editions. These, together with our own observations while teaching at the University of Texas and IBM, have prodded us to produce this third edition, which is based on the alternate edition. Our basic procedure was to reorganize and rewrite the material in each chapter, to bring some of the older material up-to-date, to improve the exercises, to add new references, and to add two new chapters — one on distributed file system and the other on the Mach operating system.

Substantive revisions were made in the following chapters

- **Chapter 2.** This is a new chapter that discusses basic computer organization. Some of the material in this chapter was previously scattered in a number of different chapters, which resulted in some confusion. New material also has been added.
- **Chapter 4.** This chapter introduces the process concept and CPU scheduling. The material in this chapter appeared in parts of old Chapters 3 and 4. We thought that the material on scheduling should precede the material on process coordination.
- **Chapter 5.** This is a new organization of the old Chapter 3. The Accent IPC example was replaced with the IPC scheme of Mach.
- **Chapter 12.** We created a separate chapter on the basic structure of distributed systems. The bulk of the material came from the old Chapter 11.
- **Chapter 13.** We created a separate chapter on distributed communication and synchronization. The bulk of the material came from the old Chapter 11.
- **Chapter 14.** This is a new chapter on distributed file system. We discuss the various ways a distributed file system can be designed and implemented. First, we discuss common concepts on which distributed file systems are based. Then, we illustrate our concepts by examining the UNIX United, NSF, Andrew, Sprite, and Locus distributed file systems.
- **Chapter 15.** This chapter discusses UNIX 4.3BSD. The material is a revision of the material on UNIX 4.2BSD previously covered in the old Chapter 12.

- **Chapter 16.** This is a new chapter on the Mach operating system. This system is designed to incorporate the many recent innovations in operating-system research to produce a fully functional, technically advanced operating system. Unlike UNIX, which was developed without regard for multiprocessing, Mach incorporates multiprocessing support throughout. Its multiprocessing support is also very flexible, ranging from shared memory systems to systems with no memory shared between processors. Mach is designed to run on computer systems ranging from one to thousands of processors. In addition, Mach is easily ported to many varied computer architectures. A key goal of Mach is to be a distributed operating system capable of functioning on heterogeneous hardware. Although there are many experimental operating systems being designed, built, and used, Mach is able to satisfy the needs of the masses better than they are because it is fully compatible with UNIX 4.3BSD. As such, it provides a unique opportunity for us to compare two functionally similar, but internally dissimilar, operating systems.

Errata

This book has benefited from the careful reading and thoughtful comments of many people in the previous editions. We have attempted to clean up every error in this new edition, but — as happens with operating systems — there will undoubtedly still be some obscure bugs. We would appreciate it if you, the reader, would notify us of any errors or omissions in the book. 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 A. Silberschatz, Department of Computer Sciences, The University of Texas at Austin, Austin, TX 78712 (e-mail — avi@cs.utexas.edu).

Acknowledgments

This book is derived from the previous editions, and so has been helped by many people, including Jeff Brumfield, Gael Buckley, Ajoy Kumar Datta, Rebecca Hartman, Richard Kieburtz, Carol Kroll, John Leggett, Michael Molloy, John Quarterman, Sara Strandtman, Charles Oualline, John Stankovic, and John Werth.

Lyn Dupré copyedited the book; Karen Shaffer edited our text into troff format. Karen Myer was very helpful with the book production.

Chapter 14 is derived from a paper by Levy and Silberschatz [1990]. John Quarterman helped us to convert the material on UNIX 4.2BSD to UNIX 4.3BSD.

We thank the following people, all of whom reviewed this edition of the book, David Black, Randy Bentson, Joseph Boykin, Wayne Hathaway, Christopher Haynes, Steven Stepanek, Louis Stevens, G. Scott Graham, Thomas LeBlanc, Robert Fowler, and Thomas Casavant.

A.S.
J.P.
P.G.

CONTENTS

PART 5 ■ Distributed Systems

CHAPTER 12 Distributed System Structures

CHAPTER 13 Distributed Coordination

CHAPTER 14 Distributed File Systems