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 change at a breathtakingly rapid rate, 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. 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 2. For code examples, we use predominantly C as well as some Java, but the reader can still understand the algorithms without a thorough knowledge of these languages.

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 operating systems, including Sun Microsystems' Solaris 2, Linux; Microsoft MS-DOS, Windows NT, and Windows 2000; DEC VMS and TOPS-20, IBM OS/2, and the Apple Macintosh Operating System.

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.

Content of this Book

The text is organized in seven major parts:

- **Overview:** Chapters 1 through 3 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 choose to skim or skip this chapter.

- **Process management:** Chapters 4 through 8 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.

- **Storage management:** Chapters 9 through 12 deal with a process in main memory 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 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 online 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.

- **I/O systems:** Chapters 13 and 14 describe the devices that attach to a computer and the multiple dimensions in which they vary. In many ways, they are also the slowest major components of the computer. Because devices differ 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 through 17 deal with a collection of processors that do not share memory or a clock—a *distributed system*. 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 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 18 and 19 explain 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 those processes that have gained proper authorization from the operating system can operate on the files, memory segments, 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 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, 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 2000, FreeBSD, Mach, and Nachos. 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 2000 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. We also cover the Nachos System, which 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. Chapter 22 briefly describes a few other influential operating systems.

The Sixth Edition

As we wrote this Sixth Edition, we were guided by the many comments and suggestions we received from readers of our previous editions, as well as by our own observations about the rapidly changing fields of operating systems and networking. We rewrote the material in most of the chapters by bringing older material up to date and removing material that was no longer of interest. We rewrote all Pascal code, used in previous editions to demonstrate certain algorithms, into C, and we included a small amount of Java as well.

We made substantive revisions and changes in organization in many of the chapters. Most importantly, we added two new chapters and reorganized the distributed systems coverage. Because networking and distributed systems have become more prevalent in operating systems, we moved some distributed systems material, client–server, in particular, out of distributed systems chapters and integrated it into earlier chapters.

- **Chapter 3, Operating-System Structures,** now includes a section discussing the Java virtual machine (JVM).

- **Chapter 4, Processes,** includes new sections describing sockets and remote procedure calls (RPCs).

- **Chapter 5, Threads,** is a new chapter that covers multithreaded computer systems. Many modern operating systems now provide features for a process to contain multiple threads of control.

- **Chapters 6 through 10** are the old Chapters 5 through 9, respectively.

- **Chapter 11, File-System Interface,** is the old Chapter 10. We have modified the chapter substantially, including the coverage of NFS from the Distributed File System chapter (Chapter 16).

- **Chapter 12 and 13** are the old Chapters 11 and 12, respectively. We have added a new section in Chapter 13, I/O Systems, covering STREAMS.
- **Chapter 14, Mass-Storage Structure,** combines old Chapters 13 and 14.
- **Chapter 15, Distributed System Structures,** combines old Chapters 15 and 16.
- **Chapter 19, Security,** is the old Chapter 20.
- **Chapter 20, The Linux System,** is the old Chapter 22, updated to cover new recent developments.
- **Chapter 21, Windows 2000,** is a new chapter.
- **Chapter 22, Historical Perspective,** is the old Chapter 24.
- **Appendix A** is the old Chapter 21 on UNIX updated to cover FreeBSD.
- **Appendix B** covers the Mach operating system.
- **Appendix C** covers the Nachos system.

The three appendices are provided online.

Teaching Supplements and Web Page

The web page for this book contains the three appendices, the set of slides that accompanies the book, in PDF and Powerpoint format, the three case studies, the most recent errata list, and a link to the authors home page. John Wiley & Sons maintains the web page at

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

To obtain restricted supplements, contact your local John Wiley & Sons sales representative. You can find your representative at the "Find a Rep?" web page: http://www.jsw-edcv.wiley.com/college/findarep

Mailing List

We provide an environment in which 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.

Suggestions

We have attempted to clean up every error in this new Edition, but—as happens with operating systems—a few obscure bugs may remain. We would appreciate hearing from you about any textual errors or omissions that you identify. If you would like to suggest improvements or to contribute exercises, we would also be glad to hear from you. Please send correspondence to Avi Silberschatz, Vice President, 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 the previous editions, the first three of which were coauthored by James Peterson. Others who helped us with previous editions include Hamid Arabnia, Randy Bentson, David Black, Joseph Boykin, Jeff Brumfield, Gael Buckley, P. C. Capon, John Carpenter, Thomas Casavant, Ajoy Kumar Datta, Joe Deck, Sudarshan K. Dhall, 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, Jim M. Ng, Banu Özden, Ed Posnak, Boris Putanec, Charles Qualline, John Quarterman, Jesse St. Laurent, John Stankovic, Adam Stauffer, Steven Stepanek, Hal Stern, Louis Stevens, Pete Thomas, David Umbaugh, Steve Vinoski, Tommy Wagner, John Werth, and J. S. Weston.

We thank the following people who contributed to this edition of the book: Bruce Hillyer reviewed and helped with the rewrite of Chapters 2, 12, 13, and 14. Mike Reiter reviewed and helped with the rewrite of Chapter 18. Parts of Chapter 14 were derived from a paper by Hillyer and Silberschatz [1996]. Parts of Chapter 17 were derived from a paper by Levy and Silberschatz [1990]. Chapter 20 was derived from an unpublished manuscript by Stephen Tweedie. Chapter 21 was derived from an unpublished manuscript by Cliff Martin. Cliff Martin helped with updating the UNIX appendix to cover FreeBSD. Mike Shapiro reviewed the Solaris information and Jim Mauro answered several Solaris-related questions.

We thank the following people who reviewed this edition of the book: Rida Bazzi, Arizona State University; Roy Campbell, University of Illinois-Chicago; Gil Carrick, University of Texas at Arlington; Richard Guy, UCLA; Max Hailperin, Gustavus Adolphus College; Ahmed Kamel, North Dakota State University; Morty Kwestel, New Jersey Institute of Technology; Gustavo Rodriguez-Rivera, Purdue University; Carolyn J. C. Schauble, Colorado State University; Thomas P. Skinner, Boston University; Yannis Smaragdakis, Georgia Tech; Larry L. Wear, California State University, Chico; James M. Westall, Clemson University; and Yang Xiang, University of Massassachusetts.

Our Acquisitions Editors, Bill Zobrist and Paul Crockett, provided expert guidance as we prepared this Edition. They were both assisted by Susannah Barr, who managed the many details of this project smoothly. Katherine Hepburn was our Marketing Manager. The Senior Production Editor was Ken Santor. The cover illustrator was Susan Cyr while the cover designer was Madelyn Lesure. Barbara Heaney was in charge of overseeing the copy-editing and Katie Habib copyedited the manuscript. The freelance proofreader was Katrina Avery; the freelance indexer was Rosemary Simpson. The Senior Illustration Coordinator was Anna Melhorn. Marilyn Turnamian helped generate figures and update the text, Instructors Manual, and slides.

Finally, we would like to add some personal notes. Avi would like to extend his gratitude to Krystyna Kwiecien, whose devoted care of his mother has given him the peace of mind he needed to focus on the writing of this book; Pete, would like to thank Harry Kasparian, and his other co-workers, who gave him the freedom to work on this project while doing his “real job”; Greg would like to acknowledge two significant achievements by his children during the period he worked on this text: Tom—age 5—learned to read, and Jay—age 2—learned to talk.

Abraham Silberschatz, Murray Hill, NJ, 2001
Peter Baer Galvin, Norton, MA, 2001
Greg Gagne, Salt Lake City, UT, 2001