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. These chapters discuss what the common features of an operating system are and what an operating system does for the user.We include coverage of both traditional PC and server operating systems, as well as operating systems for mobile devices. 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, as well as an examination of issues related to multicore systems and parallel programming.
- **Memory management**. Chapters 8 and 9 deal with the management of main memory 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 inmemory. There aremanydifferent 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 mass storage, the file system, 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. We describe the classic internal algorithms and structures of storage management and provide a firm practical understanding of the algorithms used—their 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 these 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

the slowest major components of the computer. Because they represent a

performance bottleneck, we also examine performance issues associated with I/O devices.

- **Protection and security**. Chapters 14 and 15 discuss the mechanisms necessary for the protection and security of computer systems. The processes in an operating system must be protected from one another's activities, and to provide such protection, we must ensure that only processes that have gained proper authorization fromthe operating system can operate on the files, memory, CPU, and other resources of the system. Protection is a mechanism for controlling the access of programs, processes, or users to computer-system resources. This mechanism must provide a means of specifying the controls to be imposed, as well as a means of enforcement. Security protects the integrity of the information stored in the system (both data and code), as well as the physical resources of the system, from unauthorized access, malicious destruction or alteration, and accidental introduction of inconsistency.
- **Advanced topics.** Chapters 16 and 17 discuss virtual machines and distributed systems. Chapter 16 is a new chapter that provides an overview of virtual machines and their relationship to contemporary operating systems. Included is an overview of the hardware and software techniques that make virtualization possible. Chapter 17 condenses and updates the three chapters on distributed computing from the previous edition. This change is meant to make it easier for instructors to cover the material in the limited time available during a semester and for students to gain an understanding of the core ideas of distributed computing more quickly.
- **Case studies**. Chapters 18 and 19 in the text, along with AppendicesA and B (which are available on (http://www.os-book.com), present detailed case studies of real operating systems, including Linux, Windows 7, FreeBSD, and Mach. Coverage of both Linux andWindows 7 are presented throughout this text; however, the case studies provide much more detail. It is especially interesting to compare and contrast the design of these two very different systems. Chapter 20 briefly describes a few other influential operating systems.

- **Chapter 1, Introduction,** includes updated coverage of multiprocessor and multicore systems, as well as a new section on kernel data structures. Additionally, the coverage of computing environments now includes mobile systems and cloud computing. We also have incorporated an overview of real-time systems.

- **Chapter 2, Operating-System Structures,** provides new coverage of user interfaces for mobile devices, including discussions of iOS and Android, and expanded coverage of Mac OS X as a type of hybrid system.
- **Chapter 3, Processes,** now includes coverage of multitasking in mobile operating systems, support for the multiprocess model in Google's Chrome web browser, and zombie and orphan processes in UNIX.
- **Chapter 4, Threads,** supplies expanded coverage of parallelism and Amdahl's law. It also provides a new section on implicit threading, including OpenMP and Apple's Grand Central Dispatch.
- **Chapter 5, Process Synchronization** (previously Chapter 6), adds a new section on mutex locks as well as coverage of synchronization using OpenMP, as well as functional languages.
- **Chapter 6, CPU Scheduling** (previously Chapter 5), contains new coverage of the Linux CFS scheduler andWindows user-mode scheduling. Coverage of real-time scheduling algorithms has also been integrated into this chapter.
- **Chapter 7, Deadlocks,** has no major changes.
- **Chapter 8, Main Memory,** includes new coverage of swapping on mobile systems and Intel 32- and 64-bit architectures. A new section discusses ARM architecture.
- **Chapter 9, Virtual Memory,** updates kernel memory management to include the Linux SLUB and SLOB memory allocators.
- **Chapter 10, Mass-Storage Structure** (previously Chapter 12), adds coverage of solid-state disks.
- **Chapter 11, File-System Interface** (previously Chapter 10), is updated with information about current technologies.
- **Chapter 12, File-System Implementation** (previously Chapter 11), is updated with coverage of current technologies.
- **Chapter 13, I/O,** updates technologies and performance numbers, expands coverage of synchronous/asynchronous and blocking/nonblocking I/O, and adds a section on vectored I/O.
- **Chapter 14, Protection,** has no major changes.
- **Chapter 15, Security,** has a revised cryptography section with modern notation and an improved explanation of various encryption methods and their uses. The chapter also includes new coverage ofWindows 7 security.
- **Chapter 16, Virtual Machines,** is a new chapter that provides an overview of virtualization and how it relates to contemporary operating systems.
- **Chapter 17, Distributed Systems,** is a new chapter that combines and updates a selection of materials from previous Chapters 16, 17, and 18.
- **Chapter 18, The Linux System** (previously Chapter 21), has been updated to cover the Linux 3.2 kernel.

- **Chapter 19, Windows 7,** is a new chapter presenting a case study of Windows 7.
- **Chapter 20, Influential Operating Systems** (previously Chapter 23), has no major changes.