

Chapter 1

Introducing Operating Systems

At a Glance

Instructor's Manual Table of Contents

- Overview
- Objectives
- Teaching Tips
- Quick Quizzes
- Class Discussion Topics
- Additional Projects
- Additional Resources
- Key Terms

Lecture Notes

Overview

Operating systems manage computer system hardware and software. This text explores what operating systems are, how they work, what they do, and why they do it. Chapter 1 begins by explaining what operating systems are and then presents the details of how they work. The chapter explains the major operating system software subsystem managers and their functions. The types of machine hardware on which operating systems run are then presented. The differences among batch, interactive, real-time, hybrid, and embedded operating systems are explained. A brief history of operating systems development is presented. The chapter concludes with a discussion on system architecture including the effect of object-oriented design trends in current operating systems.

Learning Objectives

After completing this chapter, the student should be able to describe:

- Innovations in operating systems development
- The basic role of an operating system
- The major operating system software subsystem managers and their functions
- The types of machine hardware on which operating systems run
- The differences among batch, interactive, real-time, hybrid, and embedded operating systems
- Multiprocessing and its impact on the evolution of operating system software
- Virtualization and core architecture trends in new operating systems

Teaching Tips

What is an Operating System?

1. Point out that every computer system consists of software (programs) and hardware.
2. Inform students that an operating system is part of a computer system in the form of software. Provide students with an overview of an operating system, pointing out its functions, such as controlling every file, every device, every section of main memory, and every nanosecond of processing time, etc. An operating system controls who may use the system and how the system is used.

Operating System Software

1. Outline all the essential managers of an operating system such as Memory Manager, Processor Manager, Device Manager, and File Manager. Note that operating systems with networking capability have a fifth essential manager called the Network Manager.
2. Discuss the importance of the User Command Interface. This communication mechanism allows users to interact directly with the operating system. Review Figure 1.1 on page 5 in the text to reinforce the abstract view of how the user interface interacts with the four essential managers in a nonnetworked environment.
3. Point out that each manager works closely with the other managers and performs its unique role. Use examples on page 6 in the text to clarify this point. There are four main tasks each manager performs:
 - Monitor its resources continuously
 - Enforce the policies that determine who gets what, when, and how much
 - Allocate the resource when it's appropriate
 - Deallocate the resource when appropriate
4. Point out how operating systems with networking capability have a fifth essential manager called the Network Manager. Discuss the advantages that it provides. Review Figure 1.2 on page 5 in the text to reinforce the abstract vision of how the user interface interacts with all five essential managers in a networked environment.

Main Memory Management

1. Describe the role of the Memory Manager and explain how it allocates and deallocates memory.

Processor Management

1. Explain how the Processor Manager keeps track of the status of each process and describe its two levels of responsibility.

Device Management

1. Discuss the role of the Device Manager, including the general process of scheduling and allocating each resource.

File Management

1. Describe the various management roles of the File Manager.
2. Discuss access control as it relates to file management.

Cooperation Issues

1. Note that each manager cannot simply perform its individual tasks on its own. Each manager must work harmoniously with every other manager. Point out that we may discuss each manager as an individual component for learning purposes; however, in reality they work together as part of the comprehensive operating system. Review Figure 1.3 on page 6 in the text to reinforce the harmonious interaction of the essential managers.
2. Using the example on page 7, review the steps each manager takes when a user types a command to execute a program.

A Brief History of Machine Hardware

1. Outline essential hardware components found in computers, explaining the basic function of each. This hardware includes main memory, also known as random access memory (RAM); the various input/output devices (I/O devices); and the central processing unit (CPU). Storage devices may be considered at this time also.
2. Provide students with an overview of the different types of system interfaces commonly found in operating systems, using Figure 1.4 on page 9 in the text.
3. Provide students with a historical overview of the development of machine hardware. Point out that until the mid-1970s, computers were classified by capacity and price. Discuss mainframes at that time in terms of size, price, and capacity.
4. Discuss the development of minicomputers, supercomputers, microcomputers, workstations, and servers. Point out their features, including capacity and applications.
5. Discuss the advances in computer technology such as dramatic changes in physical size, cost, and memory capacity. Point out that networking has become an integral part of modern computer systems.
6. Point out various handheld devices that have become popular after the rapid development of the Web. Discuss their capabilities. Note the change in operating system classification; the emphasis has turned to processor capacity due to the advances in CPU speeds. Mention that computing power increases exponentially, a principle known as Moore's Law.
7. Point out several popular operating systems and the platforms they can manage, using Table 1.1 in the text.

**Teaching
Tip**

For additional information on the basics of computer hardware, refer students to the following Web site: www.infosyssec.net/infosyssec/comphard.htm.

Types of Operating Systems

1. Outline different categories for operating systems, such as batch, interactive, real-time, hybrid, and embedded systems. Mention that each category is distinguished by response time and how data is entered into the system.
2. Provide students with an overview of batch systems. Point out that such systems in the past relied on punched cards or tape for input, and the efficiency was measured in throughput.
3. Discuss the characteristics of interactive systems. Note that their speed is faster than batch systems, but slower than real-time systems. They were introduced to assist programmers in debugging programs more interactively to provide faster turn-around time.
4. Discuss the real-time systems, providing some examples. Be sure to mention that a real-time system must be 100 percent responsive, 100 percent of the time.
5. Provide students with an overview of hybrid systems. Point out that a hybrid system takes advantage of the free time between demands for processing to execute programs that need no significant operator assistance.
6. Provide students with an overview of embedded systems, which are computers placed inside other products to add features and capabilities. Provide various examples such as embedded computers in household appliances, automobiles, digital music players, elevators, and pacemakers.
7. Discuss the requirements of operating systems for use in embedded computers. For example, operating systems with a small kernel (the core portion of the software) and with other functions that can be mixed and matched to meet the embedded system requirements will have potential in this market.

Teaching Tip

For additional information on real-time operating systems, refer to the following Web site: www.algonet.se/~staffann/developer/rtbasics.htm.

Brief History of Operating Systems Development

1. Point out that the evolution of operating system software parallels the evolution of the computer hardware they were designed to control.

1940s

1. Provide students with an overview of first-generation computers, which were based on vacuum tube technology. Point out that there was no standard operating system software at that time.

2. Mention that a typical program included every instruction needed by the computer to perform the tasks requested, and the machines were poorly utilized, i.e., the CPU processed data and made calculations for only a fraction of the available time. Basically, early programs were designed to use the resources conservatively at the expense of understandability.
3. Review Figure 1.7 on page 14 in the text explaining the novelty of where the term computer “bug” originated.

1950s

1. Provide students with an overview of second-generation computers (1955-1965). Point out that business environments placed importance on cost effectiveness; however, computers were still very expensive.
2. Outline two major improvements that were widely adopted: computer operators were hired to facilitate each machine’s operation, and job scheduling was instituted.
3. Use examples to show how job scheduling helps improve productivity. Point out that job scheduling introduced the need for control cards, which defined the exact nature of each program and its requirements. Discuss job control language (JCL).
4. Discuss various factors that helped improve the performance of the CPU, such as an increase in the speed of I/O devices, the development of access methods, the introduction of buffers and timer interrupts, etc.
5. Point out that during this time, techniques were developed to manage program libraries, create and maintain data files and indexes, randomize direct access addresses, and create and check file labels. However, programs were still run in serial batch mode, one at a time.

1960s

1. Provide students with an overview of third-generation computers dated from the mid-1960s. Point out that they were designed with faster CPUs, but their speed caused problems when they interacted with the relatively slow I/O devices.
2. Explain how the concept of multiprogramming helped solve this problem. Discuss the mechanism of its implementation.
3. Use examples to explain the concepts of passive multiprogramming and active multiprogramming. Point out the disadvantages of passive multiprogramming and how these were overcome by active multiprogramming.
4. Point out that in this generation, few advances were made in data management, and the total operating system was customized to suit users’ needs.

1970s

1. Point out that during the late 1970s, computers had faster CPUs, thus creating an even greater disparity between their rapid processing speed and slower I/O access time. Multiprogramming schemes to increase CPU use were limited by physical capacity of main memory.
2. Discuss how the concept of virtual memory solved the physical limitation issue.
3. Point out some other important developments during this time: database management software became a popular tool; a number of query systems were introduced; and programs started using English-like words, modular structures, and standard operations.
4. Review Figure 1.9 on page 19 in the text showing the massive Cray I supercomputer introduced in 1976. Discuss the implications this supercomputer had on computer systems and operating system design.

1980s

1. Discuss the various developments in the 1980s, such as improved cost/performance ratio of computer components, greater flexibility of hardware, and the introduction of the concept of firmware, etc.
2. Provide students with an overview of multiprocessing, which was introduced during this time and allowed the parallel execution of programs.
3. Point out that the evolution of personal computers and high-speed communications sparked the move to distributed processing and networked systems, enabling users in remote locations to share hardware and software resources.
4. Provide students with an overview of network operating systems and distributed operating systems.

1990s

1. Point out that the demand for Internet capability in the mid-1990s sparked the proliferation of networking capability. The World Wide Web, conceived by Tim Berners-Lee, made the Internet accessible by computer users worldwide.
2. Be sure to note that increased networking also created increased demand for tighter security to protect hardware and software.
3. Point out that the decade also introduced a proliferation of multimedia applications demanding additional power, flexibility, and device compatibility for most operating systems.

4. Review Figure 1.10 on page 21 in the text, displaying an example page from the notes of Tim Berners-Lee, the person who is credited with designing the first World Wide Web server. Discuss the implications of this design on today's computing environments.

2000s

1. Point out that primary design features of current operating systems are based on providing support for various services such as multimedia, Internet and Web access, and client/server computing, etc. Outline various requirements of computer systems in order to meet these demands, such as increased CPU speed, high-speed network attachments, and increased number and variety of storage devices.
2. Explain the process of virtualization.
3. Discuss recent advances in processing speeds.

Teaching Tip	For additional information on the history of operating systems, refer to the following Web site: www.osdata.com/kind/history.htm .
---------------------	---

Quick Quiz 1

1. Which of the following are essential managers of every operating system? (Choose all that apply.)
 - a. Memory manager
 - b. System manager
 - c. Directory manager
 - d. Process managerAnswer: a and d
2. The _____ was developed in the late 1970s to offer inexpensive computational capability to individual users.
Answer: virtual memory
3. _____ are computers placed inside other products to add features and capabilities.
Answer: Embedded systems

Threads

1. Explain the concepts of process and thread. Explain the difference between a heavyweight process and a lightweight process. Discuss how implementing threads results in maximizing the system's efficiency and satisfying the users' needs.

2. Discuss how multithreading is used to increase responsiveness in a time-sharing system to increase resource sharing and to decrease overhead.

Object-Oriented Design

1. Discuss various improvements in the system architecture of current operating systems, such as the use of object-oriented design and the reorganization of the operating system's kernel and the advantages.
2. Point out the differences between the kernel of early operating systems and the kernel of current operating systems. Explain how the kernel of current operating systems makes it easier to add new components or modify existing ones, which is useful for distributed systems.
3. Explain the concept of object-oriented design using Figure 1.13 on page 25 in the text. Point out that by working on objects, programmers can modify and customize pieces of an operating system without disrupting the integrity of the remainder of the system.

Teaching Tip	For additional information on the popular networking operating systems, refer to the following Web site: http://compnetworking.about.com/cs/networknos/ .
---------------------	--

Quick Quiz 2

1. With _____, different operating systems can run on a single computer.
Answer: virtualization
2. Which of the following is a portion of the program that can run independently of other programs?
 - a. Kernel
 - b. Thread
 - c. Firmware
 - d. PartAnswer: b
3. The _____ is the part of the operating system that resides in memory at all times, performs the most essential operating system tasks, and is protected by hardware from user tampering.
Answer: kernel

Class Discussion Topics

1. Ask students if they have worked with object-oriented programming in the past. What were some of the challenges that they faced in attempting to reuse objects in different applications? Do they think that an object-oriented approach makes software development groups more productive? Why or why not?
2. Have students discuss their feelings about the primary design features of current operating systems. Do they think that the design of operating systems based on providing support for multimedia applications or access to the Internet has resulted in increased security risks? Why or why not? Ask students to discuss some security measures that would be taken to protect hardware and software.

Additional Projects

1. Ask students to research the requirements of operating systems for use in embedded computers and write a one- to two-page report on their findings.
2. Have students research examples where multithreading is used to increase responsiveness in a time-sharing system to increase resource sharing and to decrease overhead. Ask them to present a brief report to class on one of the examples.

Additional Resources

1. Microsoft.com:
www.microsoft.com
2. IBM.com:
www.ibm.com
3. A History of Operating Systems:
www.osdata.com/kind/history.htm
4. Windows Products and Technologies History:
www.microsoft.com/windows/WinHistoryIntro.msp
5. History of UNIX:
www.bell-labs.com/history/unix/tutorial.html
6. A History of Apple's Operating System:
www.kernelthread.com/mac/oshistory/

Key Terms

- **Batch system:** a type of system developed for the earliest computers that used punched cards or tape for input, which were entered in a “batch.”
- **Central processing unit (CPU):** the component with the circuitry, the “chips,” to control the interpretation and execution of instructions.
- **Device Manager:** the section of the operating system responsible for controlling the use of devices. It monitors every device, channel, and control unit and chooses the most efficient way to allocate all of the system’s devices.
- **Embedded system:** a dedicated computer system, often small and fast, that resides in a larger physical system such as a jet aircraft or ships.
- **File Manager:** the section of the operating system responsible for controlling the use of files.
- **Firmware:** software instructions or data that are stored in a fixed or “firm” way, usually implemented on read only memory (ROM).
- **Hardware:** the physical machine and its components, including main memory, I/O devices, I/O channels, direct access storage devices, and the central processing unit.
- **Hybrid system:** a computer system that supports both batch and interactive processes.
- **Interactive system:** a system that allows each user to interact directly with the operating system via commands entered from a keyboard.
- **Kernel:** the primary part of the operating system that remains in random access memory (RAM) and is charged with performing the system’s most essential tasks, such as managing main memory and disk access.
- **Main memory:** the memory unit that works directly with the CPU and in which the data and instructions must reside in order to be processed. Also called primary storage or internal memory.
- **Mainframe:** the historical name given to a large computer system characterized by its large size, high cost, and high performance.
- **Memory Manager:** the section of the operating system responsible for controlling the use of memory. It checks the validity of each request for memory space and, if it’s a legal request, allocates the amount needed to execute the job.
- **Microcomputer:** a small computer equipped with all the hardware and software necessary to perform one or more tasks.
- **Minicomputer:** a small to medium-sized computer system, also called a midrange computer.
- **Multiprocessing:** when two or more CPUs share the same main memory, most I/O devices, and the same control program routines. They service the same job stream and execute distinct processing programs concurrently.
- **Multiprogramming:** a technique that allows a single processor to process several programs residing simultaneously in main memory and interleaving their execution by overlapping I/O requests with CPU requests.
- **Network:** a system of interconnected computer systems and peripheral devices that exchange information with one another.
- **Object-oriented:** a programming philosophy whereby programs consist of self-contained, reusable modules called objects, each of which supports a specific function, but which are categorized into classes of objects that share the same function.
- **Operating system:** the software that manages all the resources of a computer system.

- **Processor Manager:** a composite of two submanagers, the Job Scheduler and the Process Scheduler, which decides how to allocate the CPU.
- **Real-time system:** a computing system used in time-critical environments that require guaranteed response times, such as navigation systems, rapid transit systems, and industrial control systems.
- **Server:** a node that provides to clients various network services, such as file retrieval, printing, or database access services.
- **Software:** a collection of programs used to perform certain tasks. Software falls into three main categories: operating system programs, compilers and assemblers, and application programs.
- **Storage:** the place where data is stored in the computer system. Primary storage is main memory. Secondary storage is nonvolatile media, such as disks and flash memory.
- **Supercomputer:** the fastest, most sophisticated computers made, used for complex calculations.
- **Thread:** a portion of a program that can run independently of other portions. Multithreaded applications programs can have several threads running at one time with the same or different priorities.
- **Virtualization:** the creation of a virtual version of hardware or software. Operating system virtualization allows a single CPU to run multiple operating system images at the same time.
- **Workstation:** a desktop computer attached to a local area network that serves as an access point to that network.