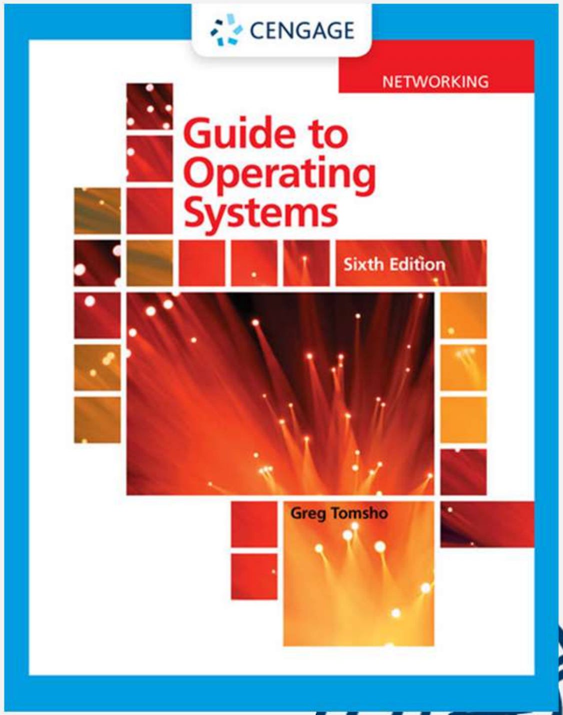
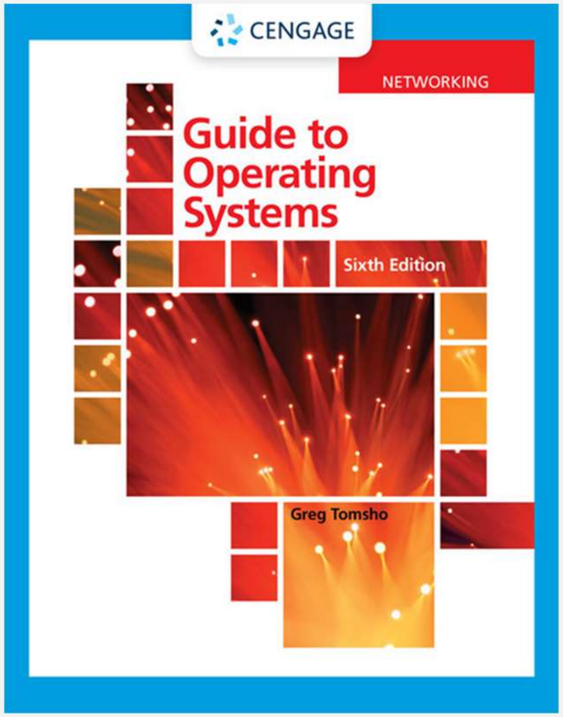
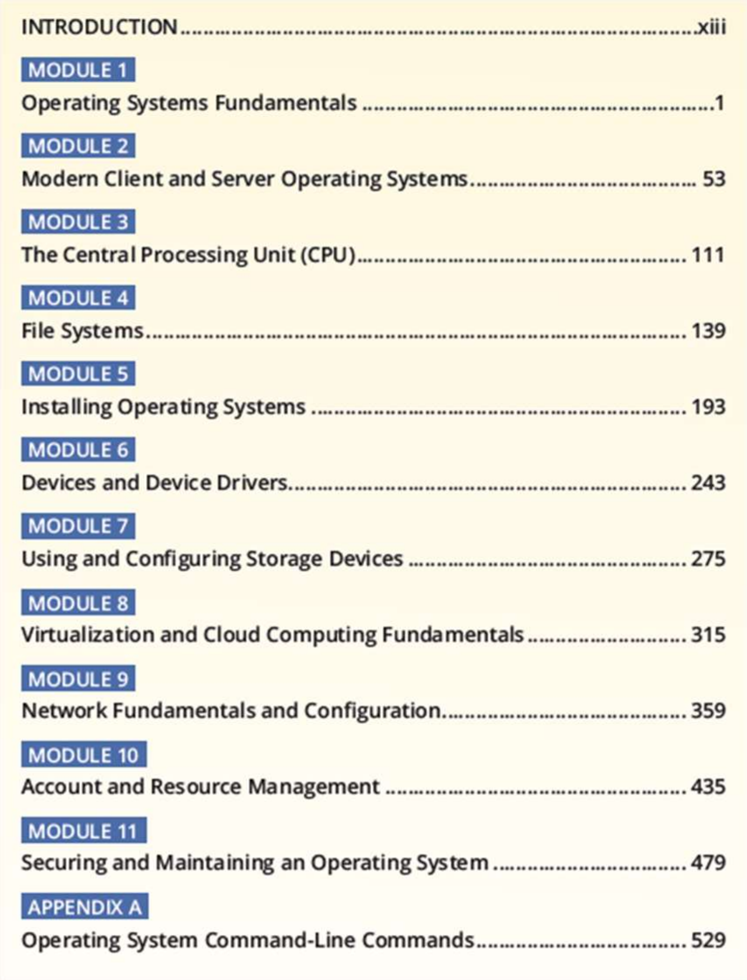
Welcome - CS 04390 Operating Systems

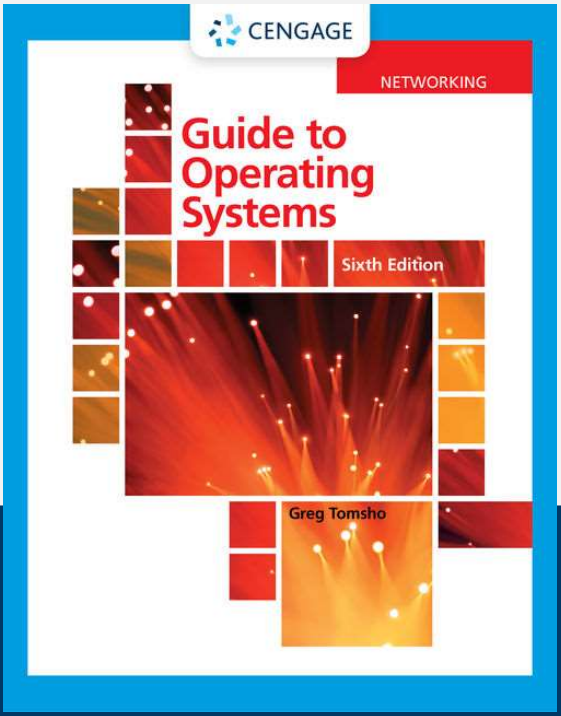
- Professor Alliegro
- New book and course – some additional modules.
- Attendance 15%
- 4 Exams – review topics before the exam
- Job interview – Why should I hire you?
- Skill assessment




CENGAGE
NETWORKING
Guide to Operating Systems
Sixth Edition
Greg Tomsho



CENGAGE



Guide to Operating Systems, 6th Edition

Module 1: Operating Systems Fundamentals

Learning Objectives

By the end of this module, you should be able to:

- Explain basic operating system concepts
- Describe the categories of operating systems
- Discuss the primary roles of an operating system
- Describe the five major operating system components
- Discuss the history of operating systems
- List several current general-purpose operating systems

An Introduction to Operating Systems (1 of 3)

- A computer’s functions and features can be broken down into three basic tasks all computers perform:
  - Input
  - Processing
  - Output
- The functions above involve some type of **computer hardware**
  - But the hardware is controlled and coordinated by the operating system (OS)

An Introduction to Operating Systems (2 of 3)

- An OS is a specialized computer program that provides the following features:
  - *User interface* – the **user interface** provides a method for users to interact with the computer
  - *Storage management* – the **file system** is a method by which an OS stores and organizes files and manages access to files
  - *Process and service management* – a **process** is a program that's loaded into memory and run by the CPU; a **service** is a type of process that runs in the background

An Introduction to Operating Systems (3 of 3)

- An OS is a specialized computer program that provides the following features (continued):
  - *Memory and I/O management* – the OS must determine if sufficient memory exists to load an application and where in memory it should be loaded
    - The OS also ensures that I/O devices such as USB ports and video cards are accessed by only one process at a time
  - *Security and resource protection* – OSs used on business systems provide methods for securing access to resources
  - *Kernel* – the **kernel** is the heart of the OS and runs with the highest priority
  - Operating Systems Introduction Video - https://youtu.be/5AjReRMoG3Y

Operating System Categories

- There are three broad categories of operating systems:
  - Single-tasking versus multitasking
  - Single-user versus multiuser
  - General-purpose versus real-time

Single-Tasking Versus Multitasking Operating Systems (1 of 4)

- A single-tasking operating system can execute only a single process at a time
  - Apple’s iOS 3 (and earlier versions) are examples
  - An **embedded system** is a computing device designed for a specific task and uses a single-tasking OS
- A multitasking operating system quickly switches between all the processes that are loaded into memory and scheduled to run

Single-Tasking Versus Multitasking Operating Systems (2 of 4)

- There are two general types of multitasking:
  - **Cooperative multitasking** – the OS gives CPU control to a process and waits for it to terminate or enter a waiting state
    - If the program does not give control back to the OS, it may hog the CPU until its operations are complete
    - No other program can run until control is given back to the OS
    - Found in early Windows versions

Single-Tasking Versus Multitasking Operating Systems (3 of 4)

- There are two general types of multitasking (continued):
  - **Preemptive multitasking** – the OS is in control of the computer at all times
    - The running process can be replaced with another process at any time based on a system interrupt, a higher-priority task requiring the CPU, or the time-slice timer expiring
    - The OS has control over how much of the computer's resources are allocated to each program
    - Computers must use more of their CPU and memory to support the OS, but the behavior of the computer is more predictable
    - The user can play music while working in documents and Web browsers

Single-Tasking Versus Multitasking Operating Systems (4 of 4)

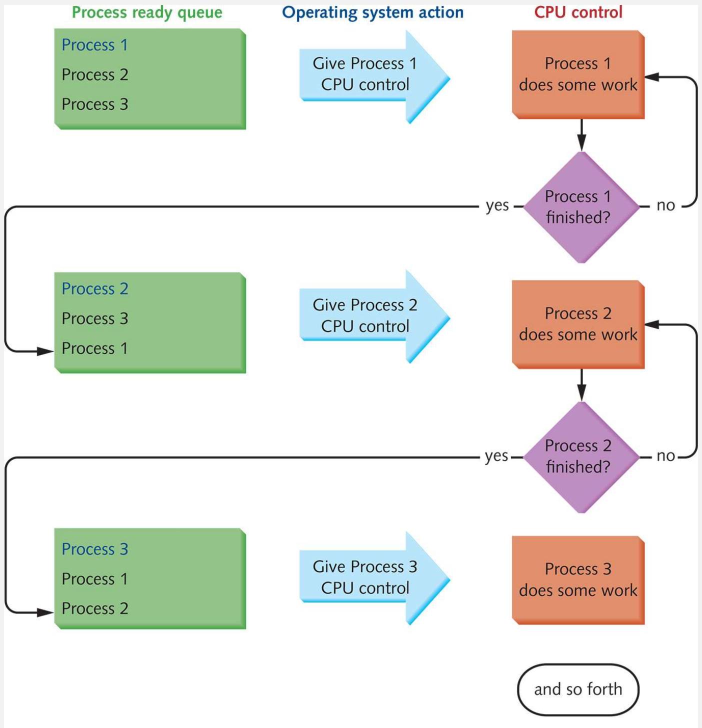


Figure 1-1 Cooperative multitasking

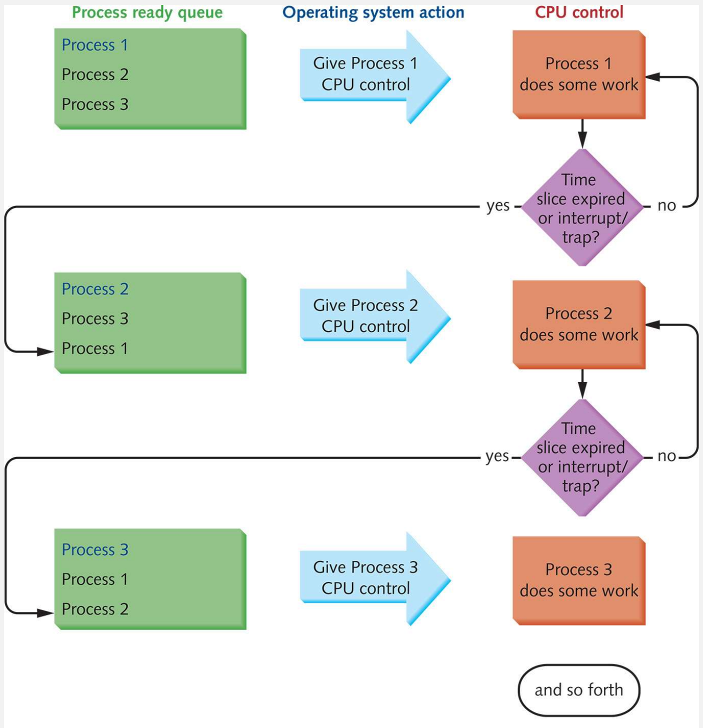


Figure 1-2 Preemptive multitasking

CENGAGE

Single-User Versus Multiuser Operating Systems

- A **single-user operating system** allows only one user at a time to interact with the OS user interface, start processes, make system configuration changes, and interact directly with the file system
  - Windows 10 and earlier Windows client versions are arguably single-user systems
- A **multiuser operating system** allows multiple users to sign in to it, start and stop processes, interact with the user interface, and access the local file system
  - Linux/UNIX OSs have always supported multiple users, and Windows Server versions also allow multiple users through Remote Desktop Protocol (RDP)

General-Purpose Versus Real-Time Operating Systems

- A **general-purpose operating system** is designed to provide a convenient user interface and run applications in a home or business environment with reasonable performance
- A **real-time operating system (RTOS)** is designed to process inputs and produce outputs in a very specific and repeatable amount of time
  - Real-time systems perform tasks like controlling assembly-line robotic systems, controlling FedEx sorting facilities, managing the electrical grid, etc.
  - Components of real-time systems are called deterministic because it is possible to determine the worst-case response times under a given set of conditions

Knowledge Check 1

- In which of the following methods does the OS give CPU control to a process and wait for it to terminate or enter a waiting state?
  - A) single-tasking
  - B) preemptive multitasking
  - C) multiuser tasking
  - D) cooperative multitasking

QUESTION

Knowledge Check 1: Answer

- In which of the following methods does the OS give CPU control to a process and wait for it to terminate or enter a waiting state?

- D) cooperative multitasking

Operating System Roles

- OSs can be defined by the role they play from the perspective of the user
- Most OSs can be classified as one of the following:
  - Client
  - Server
  - Standalone

Client Operating Systems

- The primary purpose of a **client operating system** is to run applications requested by a user and request data from network servers as necessary
  - Client OSs are usually installed on desktop or laptop computers and mobile devices and use a graphical user interface (GUI)
  - Client OSs usually include client versions of many network applications
    - For example, Windows 10 comes with Client for Microsoft Networks for accessing shared files and printers
  - Most client OSs are performance-tuned for running foreground applications (user applications) instead of background services

Server Operating Systems (1 of 3)

- The primary purpose of a **server operating system** is to share network resources and provide network services to clients
- Examples of server OSs include Windows Server, macOS Server, UNIX, and many Linux distributions
- Server OSs usually come with a DHCP server, a DNS server, a file server protocol, a Web server, and perhaps a directory service such as Microsoft’s Active Directory

Server Operating Systems (2 of 3)

- Server OSs often have built-in functions for the following:
  - **Fault tolerance** is the ability of a system to recover from a hardware or software failure
  - A server OS typically supports **clustering**, which is the ability of two or more servers to act as one
  - A **load-sharing cluster** has two or more servers that alternate in taking client requests
  - Many servers also support **hot swapping** of components such as hard disks, memory, or even processors so that if a computer fails, a new component can be swapped
  - For **scalability**, some servers support **hot adding** of components so that more storage, RAM, or processors can be added while the system is running

HPE Server and Rack

Server Operating Systems (3 of 3)

- A server OS is usually installed on a more powerful computer that may have several high-speed network interfaces and terabytes of storage
- The hardware of a server must support fault-tolerant features such as hot swapping and failover clustering
  - Server hardware can take different forms, including traditional tower computers, rack-mounted servers, and blade servers

Standalone Operating Systems

- A standalone operating system is not designed to interact with other OSs or access network resources
- Most standalone OSs run on embedded systems
- OSs that run older or isolated traffic signal systems are likely to be standalone
  - As are the OSs that run automobile electronic systems and industrial machinery
- This is changing as computer-controlled devices are being built with IoT in mind
  - So operational and performance data from these devices can be collected and analyzed to make processes more efficient

Primary Operating System Components

- This section will examine the following OS components:
  - The boot procedure
  - Process and service management
  - Memory and I/O management
  - Security and resource protection
  - The kernel

The Boot Procedure (1 of 4)

- The **boot procedure** is a series of steps performed by every computing device that starts with power to the CPU and other hardware components and ends with a running system ready to perform work
- The boot procedure can be broken down into six general steps:
  - 1. Power is applied to the CPU and other hardware components
  - 2. The CPU starts
  - 3. The CPU executes the firmware startup routines
  - 4. The operating system is located by the firmware
  - 5. The OS is loaded into RAM
  - 6. OS processes and services are started

The Boot Procedure (2 of 4)

- **Firmware** is program code that is stored on computer hardware in non-volatile memory and is responsible for performing diagnostic tests and booting the OS
- **Non-volatile memory** is memory that maintains its contents when no power is applied to the system
  - It is typically flash memory in modern systems or electrically erasable programmable read-only memory (EEPROM) in older systems
- Firmware is called the BIOS or UEFI in newer systems
- Firmware contains an initial **bootstrap** program that locates and then loads the **bootloader** program, which is responsible for loading the initial process that makes up the OS

The Boot Procedure (3 of 4)

deals with large storage devices (graphical interface + mouse support)

sometimes this is encrypted to prevent viruses

- The **basic input/output system (BIOS)** and **Unified Extensible Firmware Interface (UEFI)** are firmware programs that play a part in the boot process
- The BIOS has the following properties and performs the following tasks:
  - Resides on a chip on the computer’s motherboard flash memory
  - Initializes I/O devices
  - Performs tests at startup, such as memory and hardware component tests called the **power-on self-test (POST)** for laptops, it happens really quickly, but for servers, it's longer
  - Provides a user interface to allow hardware configuration such as CPU overclocking or disk configuration
  - Locates and loads the OS that interfaces with the user

The Boot Procedure (4 of 4)

- BIOS configuration is stored in a **complementary metal oxide semiconductor (CMOS)** memory chip
- Users can make changes to the BIOS configuration by accessing the BIOS setup screen before the computer boots
- When a PC is turned on:
  - The machine wakes up and begins executing a startup program inside the BIOS
  - This program initializes the screen and keyboard, tests hardware, and initializes the hard disk and other drives
  - The main operating system is loaded onto the hard disk
  - BIOS Video https://youtu.be/zIYkol851dU

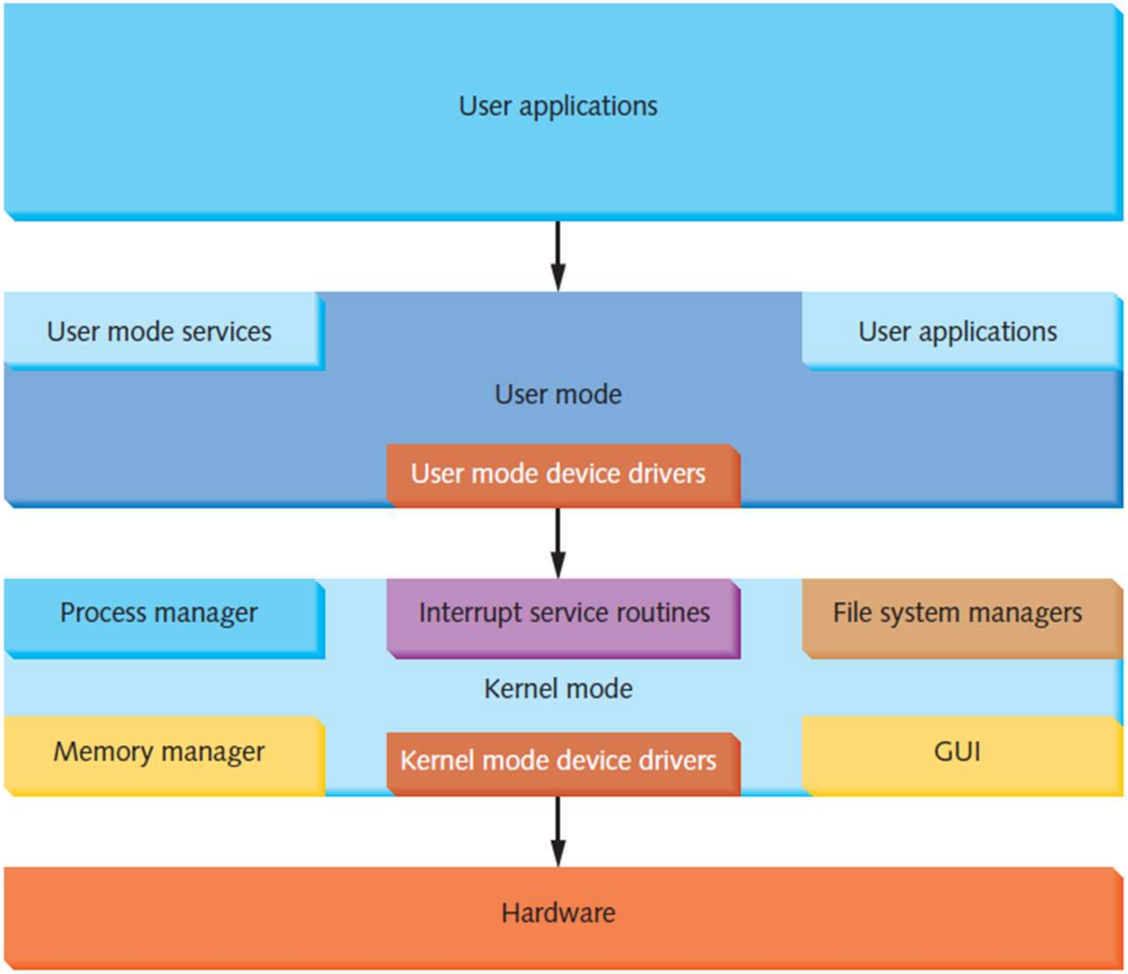


Figure 1-7 General operating system design

an application is a process the user interacts with

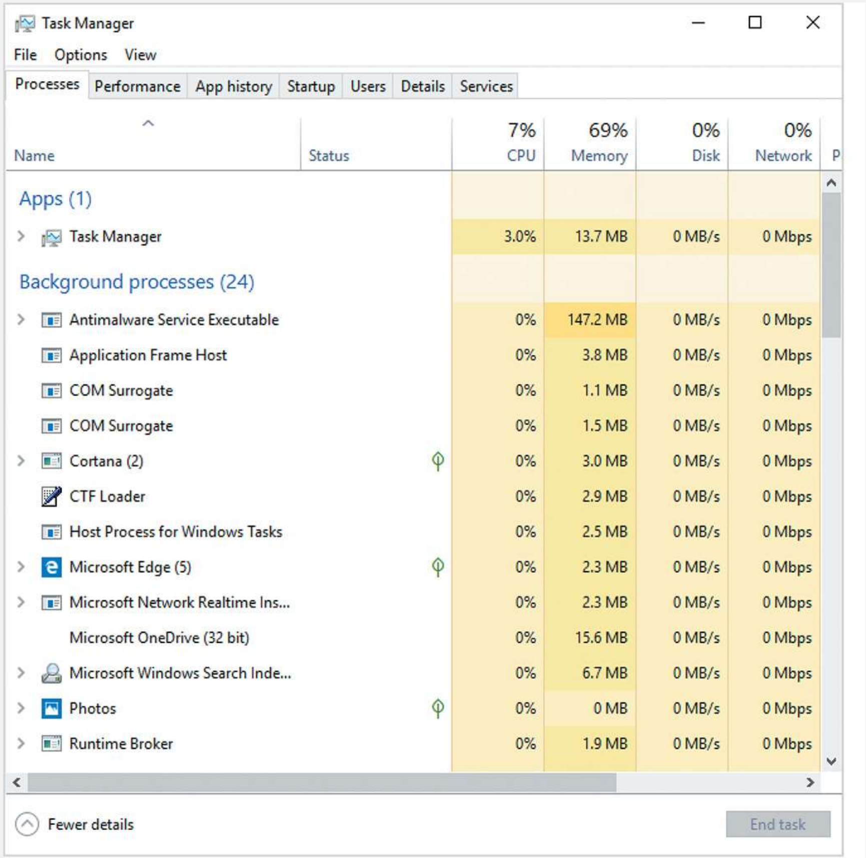
Process and Service Management (1 of 4)

meanwhile, a service is a process that runs in the background/doesn't interact with the desktop

anything that's running

- A process is a program that is loaded into memory and executed by the CPU
- A program can be a user application, an OS service, or even the kernel
- On a Windows 10 system, you can see all running processes by using Task Manager
- Background processes listed by Task Manager are not essential for the OS to function
- Windows processes are required for full system functionality and include the registry service, user interface components, network services, interrupt handlers, and logon services

Process and Service Management (2 of 4)


Task Manager
File Options View
Processes Performance App history Startup Users Details Services
Fewer details
End task

Name	Status	7% CPU	69% Memory	0% Disk	0% Network
Apps (1)					
Task Manager		3.0%	13.7 MB	0 MB/s	0 Mbps
Background processes (24)					
Antimalware Service Executable		0%	147.2 MB	0 MB/s	0 Mbps
Application Frame Host		0%	3.8 MB	0 MB/s	0 Mbps
COM Surrogate		0%	1.1 MB	0 MB/s	0 Mbps
COM Surrogate		0%	1.5 MB	0 MB/s	0 Mbps
Cortana (2)		0%	3.0 MB	0 MB/s	0 Mbps
CTF Loader		0%	2.9 MB	0 MB/s	0 Mbps
Host Process for Windows Tasks		0%	2.5 MB	0 MB/s	0 Mbps
Microsoft Edge (5)		0%	2.3 MB	0 MB/s	0 Mbps
Microsoft Network Realtime Ins...		0%	2.3 MB	0 MB/s	0 Mbps
Microsoft OneDrive (32 bit)		0%	15.6 MB	0 MB/s	0 Mbps
Microsoft Windows Search Inde...		0%	6.7 MB	0 MB/s	0 Mbps
Photos		0%	0 MB	0 MB/s	0 Mbps
Runtime Broker		0%	1.9 MB	0 MB/s	0 Mbps

Figure 1-8 A list of processes on a Windows 10 system

Process and Service Management (3 of 4)

parent vs child relationship

- Kernel Mode Versus User Mode Processes

closest connection to CPU

  - **Kernel mode** is a CPU mode in which a process has unrestricted access to the computer hardware and has access to privileged CPU instructions
    - Privileged CPU instructions include those that switch between kernel and user mode and those that configure interrupts, timers, and I/O devices
  - **User mode** is a CPU mode in which the process can only access memory locations allocated to it by the OS, must ask the OS to access I/O devices on its behalf, and can’t execute privileged CPU instructions

CENGAGE

Process and Service Management (4 of 4)

- Process Creation and Termination

  shown in process explorer in windows

  - When a process is created, it is assigned a **process ID (PID),** which allows other processes to communicate with it and the user to manage it
  - A **file handle** is an identifier for a file that contains all the information the file system needs to locate and access the file
  - Processes are usually created in one of four ways:
    - *By the bootloader*
    - *Through a request by the operating system* services
    - *By another process*
    - *By a user*

in task manager, performance tab, resource monitor can show pid

response time recommended < 15 ms

a swap file makes physical memory in the hard drive (slows it down, particularly spinning disk)

Memory and I/O Management (1 of 5)

- Memory is managed by a dedicated memory manager that runs in kernel mode
- When a process is loaded and started, the OS must allocate enough memory for the process and its data
  - When a process terminates, the OS must mark the memory as free so other processes can use it
- I/O management is required to prevent more than one process from accessing an I/O device at the same time

Memory and I/O Management (2 of 5)

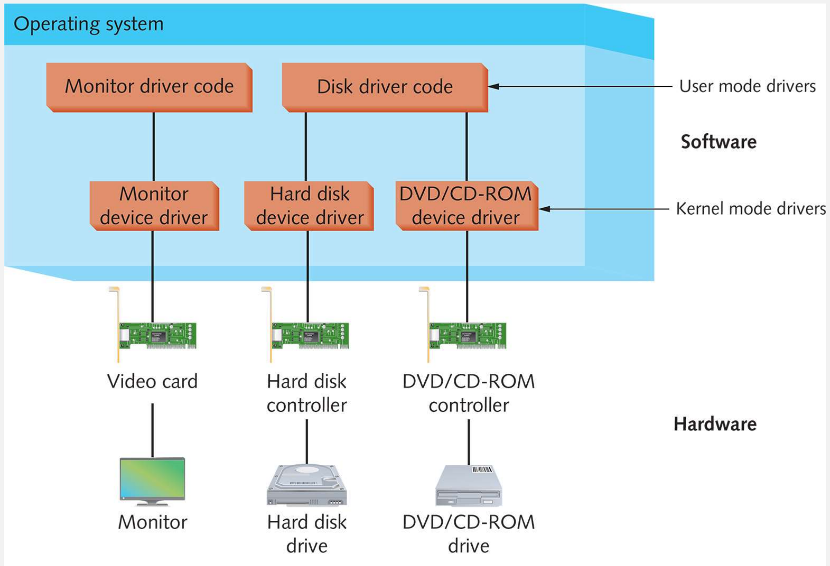


Figure 1-12 Device drivers provide communication between the OS and hardware devices

- Access to all devices is done in an orderly fashion, through a single process called a device driver
  - A **device driver** is usually a kernel mode process that accesses hardware devices directly on behalf of user mode processes or other kernel mode processes

Memory and I/O Management (3 of 5)

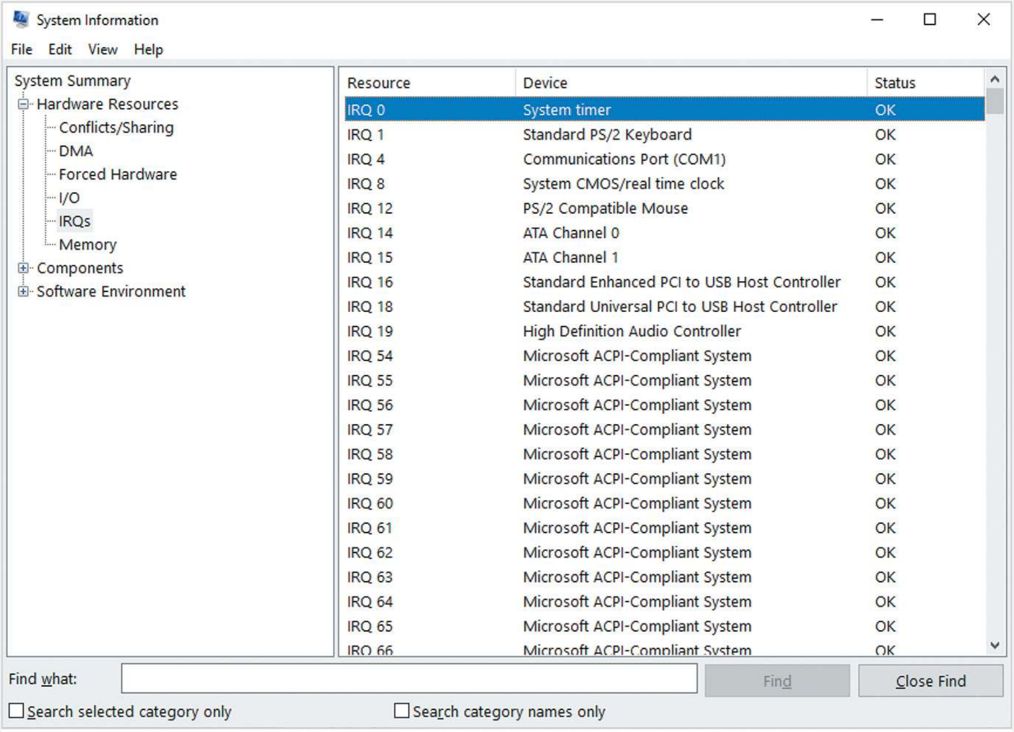
- Interrupts
  - gives time but doesn't release it back to the system
  - **Polling** is a process where the OS checks each device one after the other in a round-robin fashion to see if it requires attention
  - Because of the shortcomings of polling, most systems use interrupts

    (ex. raising your hand and doesn't always call on it)
  - An **interrupt** is a signal, usually generated by an I/O device, that alerts the CPU and OS that a device needs attention
  - Pins on a CPU are usually called **interrupt request (IRQ) lines**
    - When the CPU detects the IRQ line, it stops what it is doing, determines the source of the interrupt, and starts execution of the **interrupt service routine (ISR),** also referred to as an interrupt handler

Memory and I/O Management (4 of 5)

- Interrupts (continued)
  - Interrupts are prioritized
    - Critical system devices such as the system timer have the highest priority
  - A software interrupt is called a **trap** and can be generated by user mode programs when they need immediate kernel services
  - To prevent unnecessary interruptions, ISRs turn off interrupts of equal or lower priority; this is called **interrupt masking**

Memory and I/O Management (5 of 5)



Resource	Device	Status
IRQ 0	System timer	OK
IRQ 1	Standard PS/2 Keyboard	OK
IRQ 4	Communications Port (COM1)	OK
IRQ 8	System CMOS/real time clock	OK
IRQ 12	PS/2 Compatible Mouse	OK
IRQ 14	ATA Channel 0	OK
IRQ 15	ATA Channel 1	OK
IRQ 16	Standard Enhanced PCI to USB Host Controller	OK
IRQ 18	Standard Universal PCI to USB Host Controller	OK
IRQ 19	High Definition Audio Controller	OK
IRQ 54	Microsoft ACPI-Compliant System	OK
IRQ 55	Microsoft ACPI-Compliant System	OK
IRQ 56	Microsoft ACPI-Compliant System	OK
IRQ 57	Microsoft ACPI-Compliant System	OK
IRQ 58	Microsoft ACPI-Compliant System	OK
IRQ 59	Microsoft ACPI-Compliant System	OK
IRQ 60	Microsoft ACPI-Compliant System	OK
IRQ 61	Microsoft ACPI-Compliant System	OK
IRQ 62	Microsoft ACPI-Compliant System	OK
IRQ 63	Microsoft ACPI-Compliant System	OK
IRQ 64	Microsoft ACPI-Compliant System	OK
IRQ 65	Microsoft ACPI-Compliant System	OK
IRQ 66	Microsoft ACPI-Compliant System	OK

Figure 1-13 Interrupts on a Windows 10 system

- Figure 1-13 shows a list of IRQs and the devices or processes using them
  - A certain class of interrupts called **non-maskable interrupts (NMI)** cannot be turned off
    - NMIs are primarily reserved for error conditions such as divide by zero or memory errors, but they can also be used for system debugging

Security and Resource Protection

- Modern general-purpose OSs and RTOSs running on embedded systems provide controls to prevent unauthorized access to the system and its data
- Malware protection starts with protected memory space
- Authentication and Authorization
  - **Authentication** verifies that an account trying to access a system is valid and has provided valid credentials (username and password)
  - **Authorization** verifies that an authenticated account has permission to perform an action on a system

The Kernel

- The kernel runs with the highest priority on the system
- Scheduling computer processes and managing resources are key tasks of the OS kernel
- Jobs performed by the kernel can include the following:
  - Managing interactions with the CPU
  - Managing interrupt handlers and device drivers
  - Handling basic computer security
  - Managing use of the computer’s memory
  - Managing priority levels assigned to programs and computer processes

Knowledge Check 2

- The ability of two or more servers to act as one is known as which of the following?
  - A) hot swapping
  - B) clustering
  - C) bootstrapping
  - D) hot adding

QUESTION

Knowledge Check 2: Answer

- The ability of two or more servers to act as one is known as which of the following?

- B) clustering

ANSWER

A Short History of Operating Systems (1 of 5)

- Initially, computers were used as large automated calculators for mathematical and statistical problems
- Legitimate use can be traced back 100 years or more but there were no practical designs used by significant numbers of people until the late 1950s
- Operating systems were rudimentary, often not able to do more than read punch cards or tape and write output to machines resembling typewriters
  - It did not take long before applications evolved to do something useful for a broader audience

vacuum here functions as a transistor

A Short History of Operating Systems (2 of 5)

- Computers of the late 60s and early 70s were crude by today's standards but were quite capable
  - Contributed to the development of space travel, submarine-based ballistic missiles, and the global financial community
- The period of growth during the 1960s and 1970s saw:
  - The beginning of the Internet
  - Creation of input and output devices such as display terminals and magnetic disks
  - Creation of more operating systems such as OS/8 and UNIX

A Short History of Operating Systems (3 of 5)

- In the mid-1960s, a simple programming language was developed at Dartmouth College
  - BASIC – Beginner's All-purpose Symbolic Instruction Code
- In 1975, Bill Gates wrote a compiler (software that turns computer code written by people into code that is understood by computers) for BASIC
  - He sold it to a company called Micro Instrumentation Telemetry Systems (MITS), which became the first company to produce a desktop computer
- Other programming languages introduced around this time included Pascal, C, and other versions of BASIC

A Short History of Operating Systems (4 of 5)

bought Q-DOS

- Bill Gates started a new company called Microsoft, which led to the development of Microsoft Disk Operating System or MS-DOS
  - IBM adopted MS-DOS, which ran on a prototype of a new microcomputer being developed by IBM called the personal computer (PC)
  - MS-DOS was the first widely distributed operating system for microcomputers that had to be loaded from disk or tape
- In 1984, Apple Macintosh developed a GUI and mouse pointing device, which allowed users to interact with the OS on a graphical screen

A Short History of Operating Systems (5 of 5)

- In 1985, Microsoft released Windows, which provided a GUI and many of the same functions as the Mac OS
- See Table 1-2 for an extensive look at operating system releases from 1968 to 1999

Activity

- Try Hands-On Project 1-1, found at the end of Module 1, to learn more about the history of computers and operating systems
- Hands-On Project 1-1: Exploring the History of the Computer
    - Description: In this project, you use the Internet to review the history of computers, software, and the Internet

Current General-Purpose Operating Systems (1 of 4)

- The most common OSs in today’s home and business computing environments fall into several families:
    - Windows client operating systems
    - Windows server operating systems
    - UNIX/Linux operating systems
    - Apple Macintosh macOS

Current General-Purpose Operating Systems (2 of 4)

- At the time of this writing, the most frequently used Windows client versions are Windows 7 and 10
  - As of January 2020, Windows 7 reached its “end of life,” which means Microsoft discontinued support for Windows 7
  - This book will focus on Windows 10
- The most popular Microsoft server operating systems are Windows Server 2012, Windows Server 2016, and Windows Server 2019
  - Support for Windows Server 2008 ended in January 2020 and support for Windows Server 2012 is scheduled to end in October 2023
  - Feature descriptions and screenshots of Windows Server in this book will be based on Windows Server 2016 and Windows Server 2019

Current General-Purpose Operating Systems (3 of 4)

- There are many flavors of UNIX, but the two main design standards are the Berkeley Software Distribution (BSD) standard and the System V Release 4 (SVR4) standard
  - This book focuses on SVR4 UNIX
- Linux OS is a UNIX look-alike system that is popular as a server OS in business, education, and government
  - The Linux screenshots and features covered in this book are based on Fedora 30 Workstation and Fedora 30 Server

Current General-Purpose Operating Systems (4 of 4)

- macOS for Apple Macintosh computers is popular in the educational and graphics sectors
  - Corporate users sometimes regard macOS as difficult to set up for networking in medium-sized to large organizations with complex networks
  - Apple has addressed many of these concerns
  - macOS is popular with home users because the desktop is intuitive and home network setup is user friendly
  - The Mac screenshots and features covered in this book are based on macOS Mojave

Summary (1 of 3)

- All computers perform three basic tasks: input, processing, and output
- An operating system (OS) provides the foundation upon which to run the components of a computer and execute applications
- A basic task of an OS is to enable a computer to perform I/O functions so that it can use software applications and communicate with computer hardware
- OSs can be understood in terms of characteristics such as time sharing, real-time operation, and multiuser capabilities
- Early OSs tended to be single-tasking, but modern systems are largely multitasking

Summary (2 of 3)

- A true multiuser system is one in which multiple users access and run a single application on a single computer at the same time
- Two common types of OSs are desktop (or client) and server OSs
- The BIOS is a low-level program code that operates between the computer hardware and a higher-level operating system to initiate communication with hardware devices
- Device drivers can extend the native function of an operating system to provide access and control over different types of devices

Summary (3 of 3)

- The history of operating systems and computers represents a progression from huge computers to large computers to desktop-sized computers that have powerful operating systems
- An OS may be geared to run a large mainframe or a small PC
- One of the most significant advances in operating systems is the refinement of the GUI in Windows-based and Mac OS systems
- Current popular OSs include Windows 10, Server 2012/R2, Server 2016, Server 2019, UNIX/Linux, and macOS Mojave