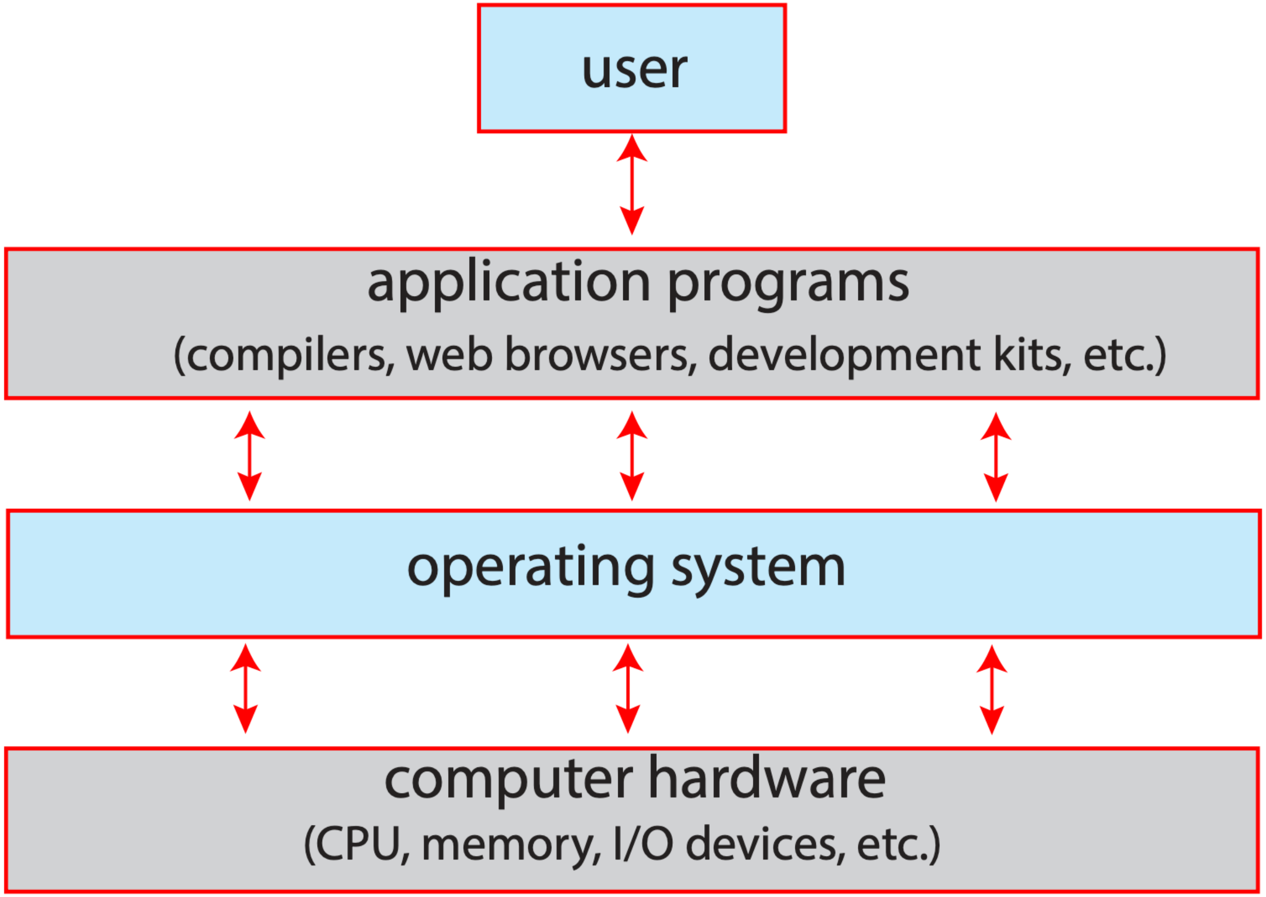
CS 3113 Introduction to Operating Systems

Topic #1. Introduction

Q1: What is a Computer System?


user
application programs
(compilers, web browsers, development kits, etc.)
operating system
computer hardware
(CPU, memory, I/O devices, etc.)

Q2: What is an Operating System?

- A program that acts as an intermediary between a "user" of a computer and the computer hardware
  - …a ***User*** can really be a person, an application program or another computer

- Some examples of operating systems:

Android
Windows 10
MICROSOFT
Microsoft Windows
ıbuntı
Ubuntu
Linux
mac OS
macOS
Chrome OS

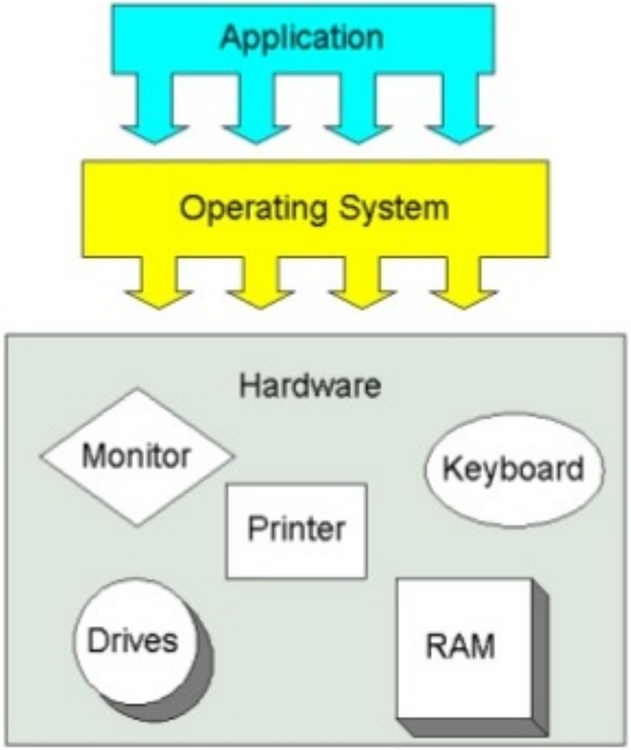
Q2: What is an Operating System? (Cont'd)

- A collection of software that manages computer hardware resources and provides common services for computer programs

- The one program running all times on computer – usually called kernel

- A vital component of the system software in a computer system

- Application programs usually require an operating system to function

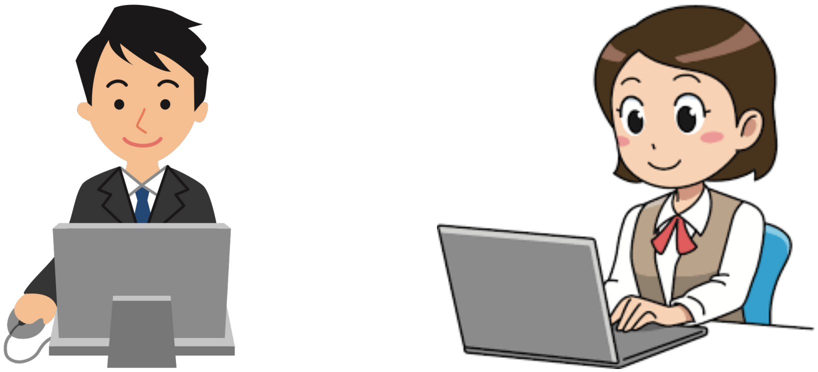


Q3: What are the Goals of an OS?

- Three major goals:
  - Execute user programs and make solving user problems easier
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner

Q4: Why This Course?

- Knowledge of how operating systems work is crucial to proper, efficient, effective, and secure programming.

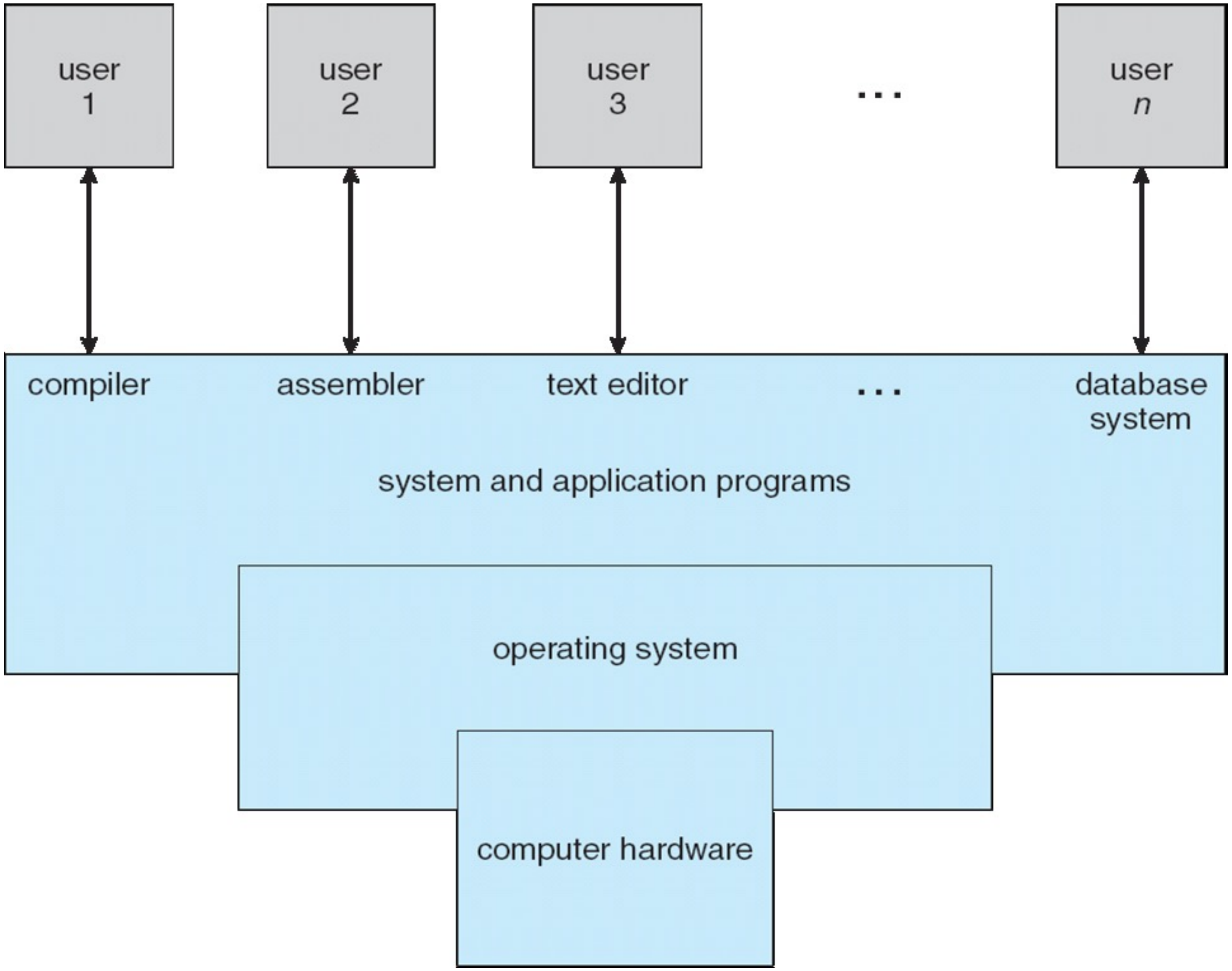


- No matter you program OSs, or write programs on OSs, or just use OSs, learning OSs will be beneficial.

Computer System Structure

- Hardware
  - provides basic computing resources
- Operating system
  - Controls and coordinates use of hardware among various applications and users
- Application programs
  - define the ways in which the system resources are used to solve the computing problems of the users
  - Examples: Word processors, compilers, web browsers, database systems, video games
- Users
  - People, machines, other computers

Computer System Structure (Cont'd)



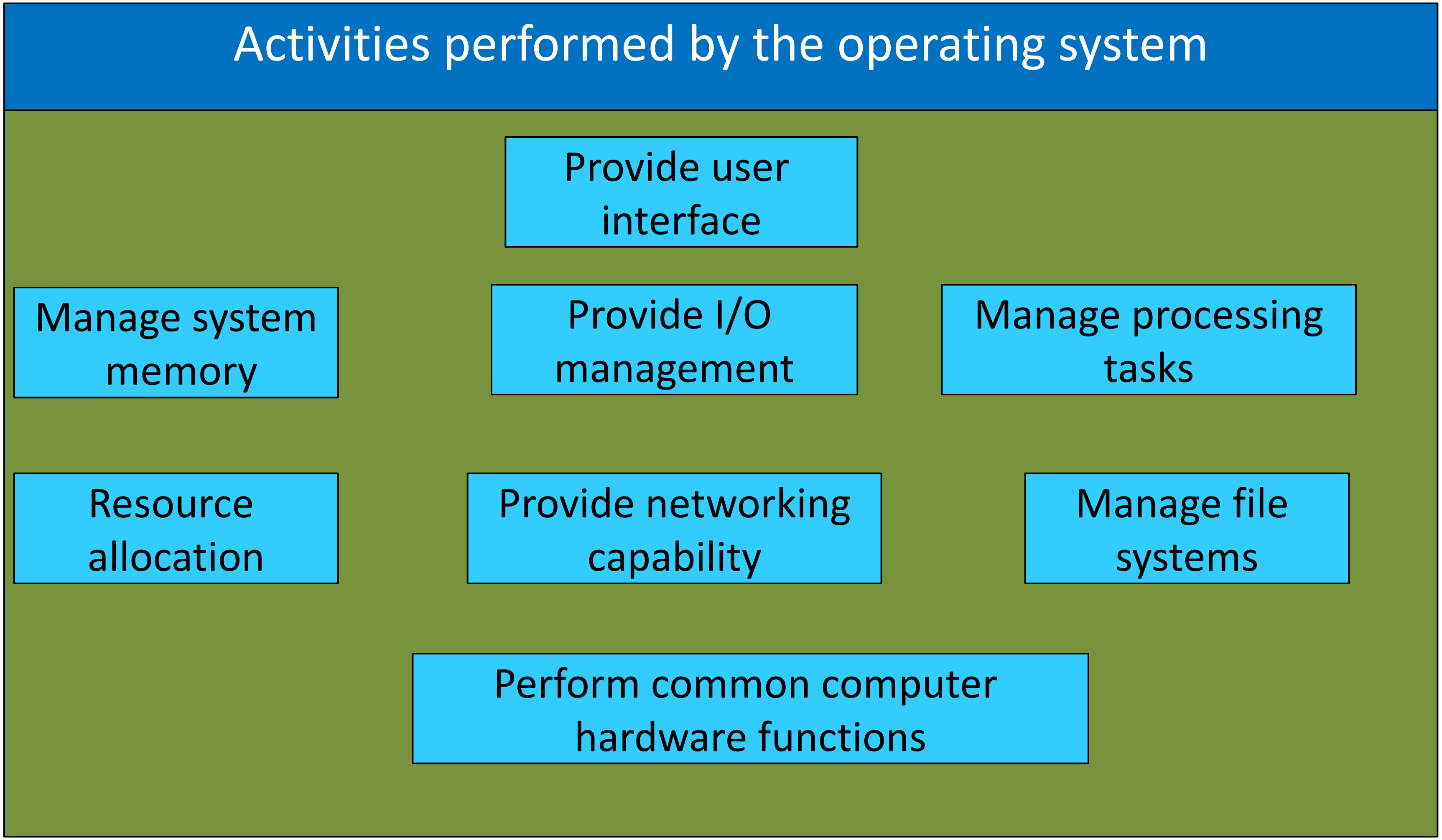
Outline

- 1.1 What Operating Systems Do
- 1.2 Computer-System Organization
- 1.3 Computer-System Architecture
- 1.4 Operating-System Operations
- 1.5 Resource Management

What Operating Systems Do?

Types of OS	OS goals
OS for Single User System	Ease of use
OS for Multi User Systems (mainframe or minicomputer)	Maximize resource utilization
Users of dedicate systems (workstations)	Compromise between individual usability and resource utilization
Handheld computers	optimized for usability and battery life
Embedded computers	Run without user intervention (with little or no user interface)

What Operating Systems Do (Cont’d)



Activities performed by the operating system

Provide user interface

Manage system memory

Provide I/O management

Manage processing tasks

Resource allocation

Provide networking capability

Manage file systems

Perform common computer hardware functions

Operating System Definition

- OS is a resource allocator
  - Manages all hardware resources
  - Decides between conflicting requests for efficient and fair resource use
- OS is a control program
  - Controls execution of programs to prevent errors and improper use of the computer
- OS provides abstractions
  - Hides the details of the hardware
  - Provides an interface that allows a consistent experience for application programs and users

Operating System Definition

What are common abstractions provided by the OS?

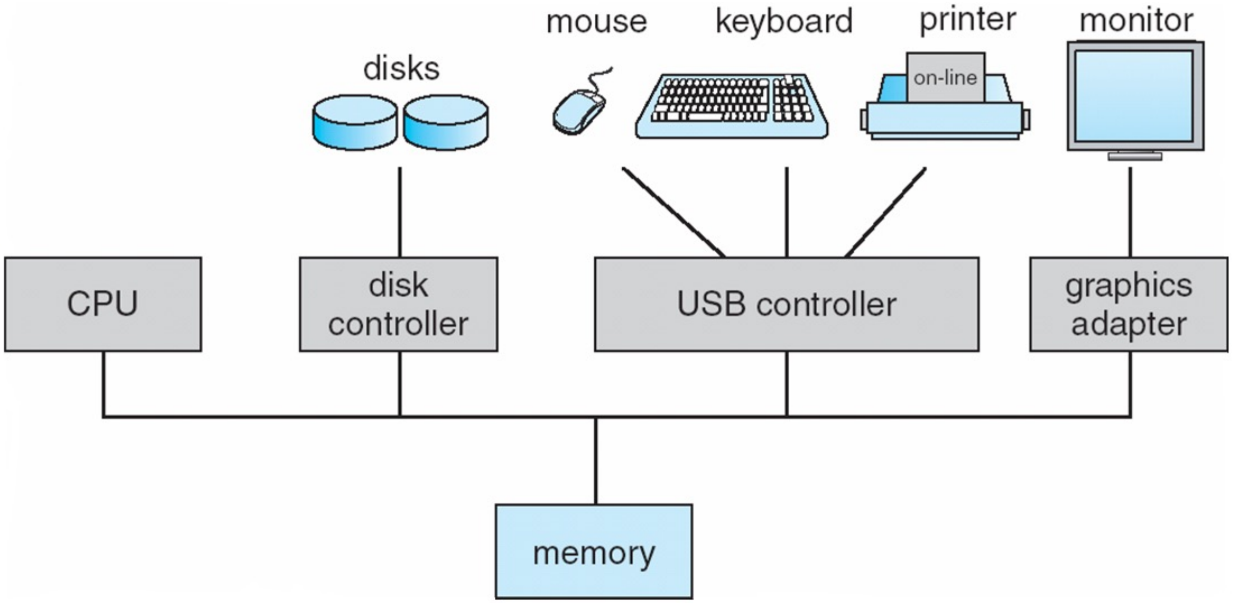
- A program has exclusive access to the CPU(s) and other hardware devices
- A program has unbounded access to memory
- Directories and files
- Reliable communication between programs and computers
- No errors in: execution, communication, device interaction

Outline

- 1.1 What Operating Systems Do
- 1.2 Computer-System Organization
- 1.3 Computer-System Architecture
- 1.4 Operating-System Operations
- 1.5 Resource Management

Computer System Organization

- A modern computer-system
  - One or more CPUs, device controllers connect through common bus providing access to shared memory
  - Concurrent execution of CPUs and devices competing for memory cycles



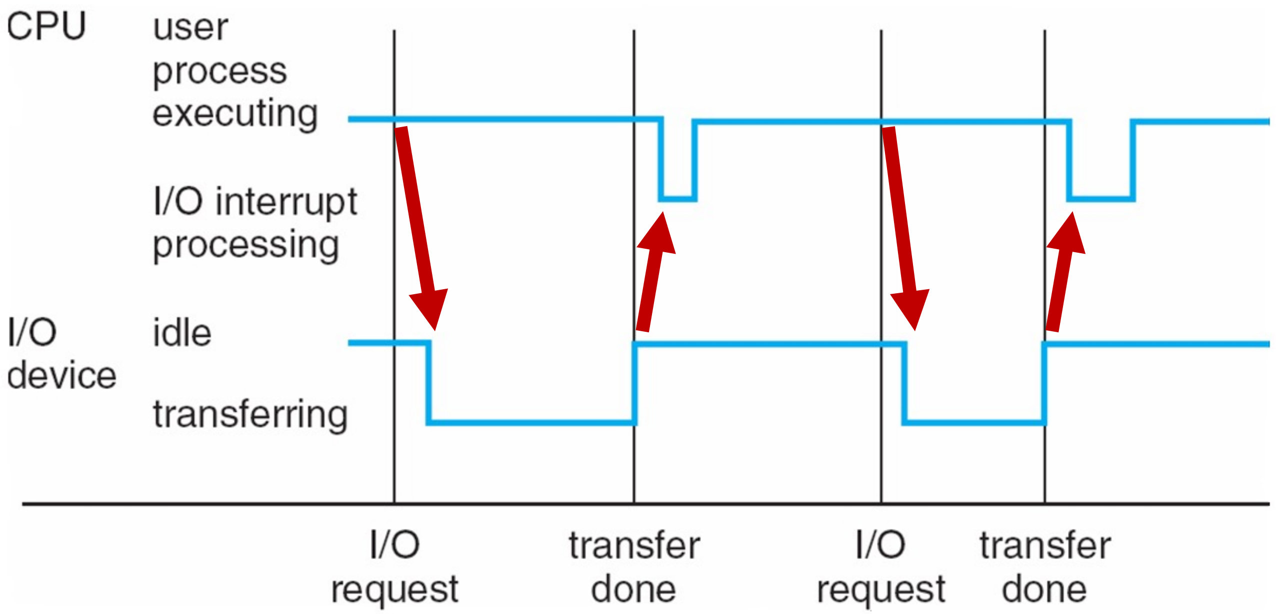
Computer-System Operation

- I/O devices and the CPU can execute concurrently
- Each device controller is in charge of a particular device type
- Data sent to or received from the device are stored in a local buffer
- CPU moves data from/to main memory to/from local buffers
- When a device controller completes an I/O operation, it informs the CPU by causing an interrupt

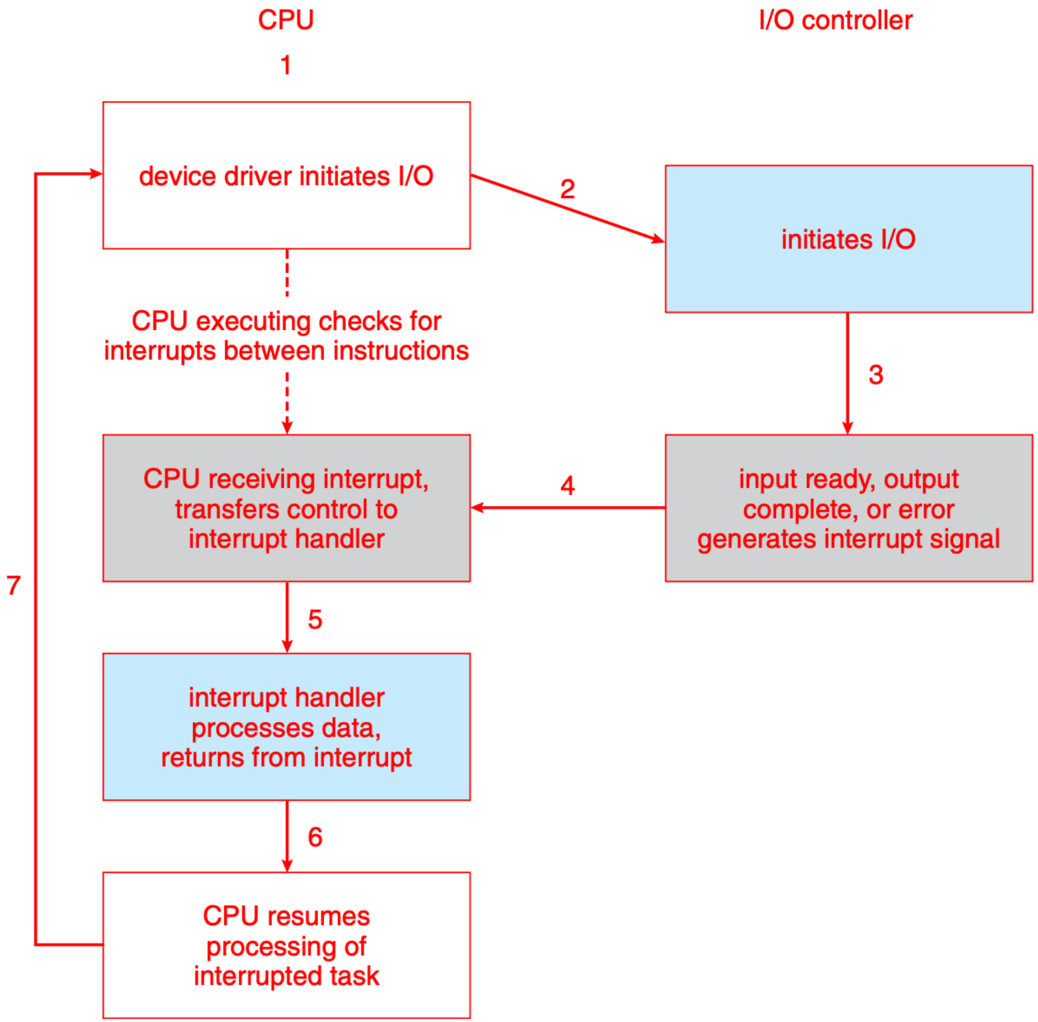
Interrupts

- An operating system is **interrupt driven**
  - An interrupt transfers control from the currently executing program to the appropriate interrupt service routine
  - Interrupt architecture must save the address of the interrupted instruction, as well as the register state
  - A **trap** or **exception** is a software-generated interrupt caused either by an error or a user request

Interrupt Timeline



Interrupt-driven I/O cycle



CPU

I/O controller

1

device driver initiates I/O

2

initiates I/O

CPU executing checks for interrupts between instructions

3

CPU receiving interrupt, transfers control to interrupt handler

4

input ready, output complete, or error generates interrupt signal

7

5

interrupt handler processes data, returns from interrupt

6

CPU resumes processing of interrupted task

I/O Structure

- User program does not have direct access to the devices (it is prevented explicitly!)
- Instead, a request for access is made to the OS through the use of a system call
  - Special function that is able to access the kernel-level data structures and I/O system
- After I/O starts, control returns to user program without waiting for I/O completion

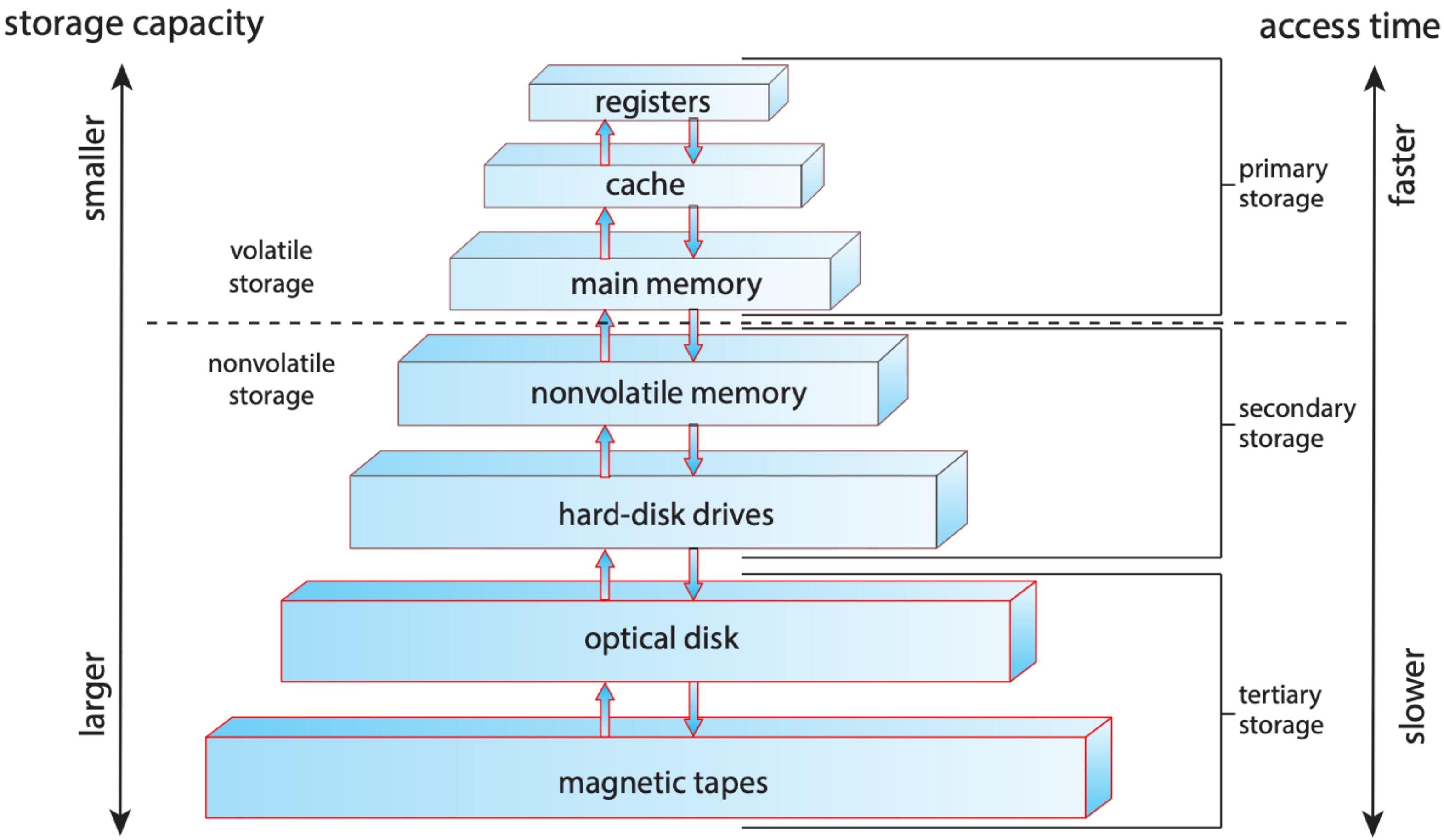
Storage Definitions

- Bit: contains a value of 0 or 1
- Byte: 8 bits. Fundamental unit of memory
- Word: multiple bytes (system dependent)
  - In modern laptops: 8 bytes
- 2^10 bytes: kilobyte
- 2^20 bytes: megabyte
- 2^30 bytes: gigabyte
- 2^40 bytes: terabyte

Storage Types (some)

- Main memory – only large storage media that the CPU can access directly
  - Random access, typically volatile
- Secondary storage – extension of main memory that provides large **nonvolatile** storage capacity
  - Hard disks – rigid metal or glass platters covered with magnetic recording material
  - Disk surface is logically divided into **tracks**, which are subdivided into **sectors**
  - The **disk controller** determines the logical interaction between the device and the computer
- Solid-state disks – faster than hard disks, nonvolatile
  - Various technologies
  - Becoming more popular

Storage Hierarchy



Storage Hierarchy

- Storage systems organized in hierarchy. Each level involves trade-offs:
  - Speed
  - Cost
  - Volatility
- **Caching** – copying information into faster storage system
  - Main memory can be viewed as a cache for secondary storage

Caching

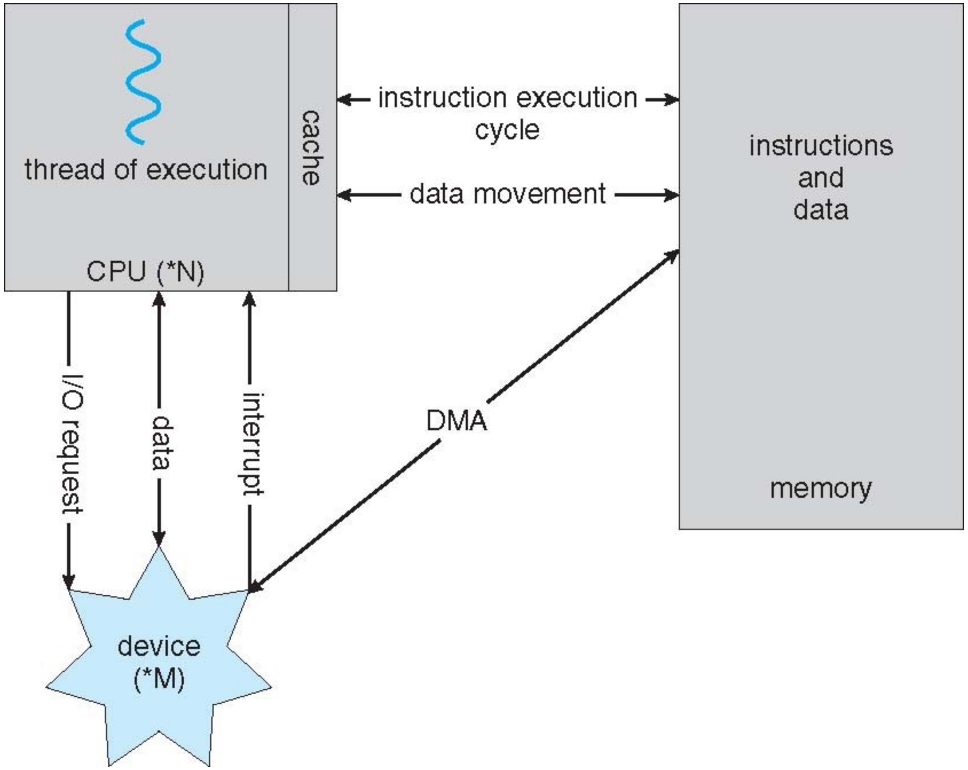
Information in use copied from slower to faster storage temporarily

- Important principle, performed at many levels in a computer (in hardware, operating system, software)
- Faster storage (cache) checked first to determine if information is there
  - If it is, information used directly from the cache (fast)
  - If not, data copied to cache and used from there
- Cache management is an important design choice
  - Including: cache size and replacement policy

Direct Memory Access

- Used for high-speed I/O devices able to transmit information at close to memory speeds
- Device controller transfers blocks of data from buffer storage directly to main memory without CPU intervention
- Only one interrupt is generated per block, rather than the one interrupt per byte

Data Flow in a Modern Computer



Outline

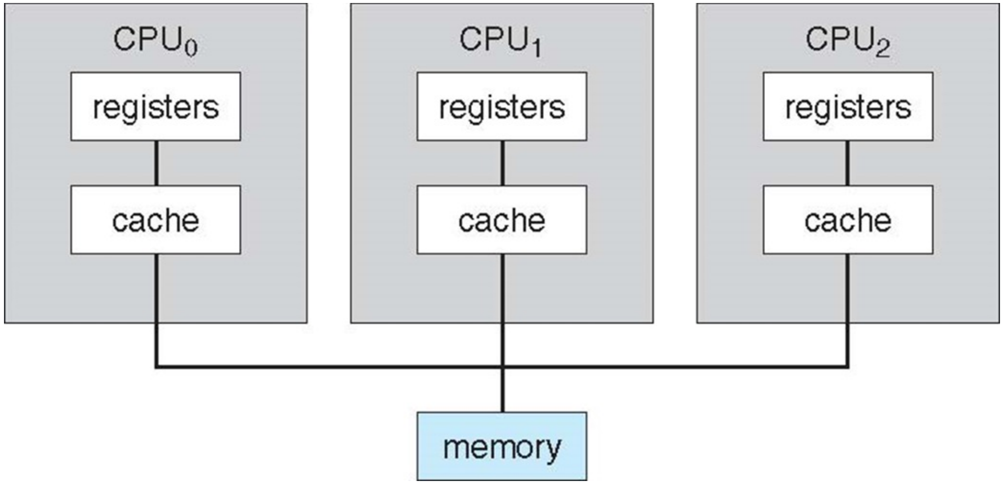
- 1.1 What Operating Systems Do
- 1.2 Computer-System Organization
- 1.3 Computer-System Architecture
- 1.4 Operating-System Operations
- 1.5 Resource Management

Computer-System Architecture

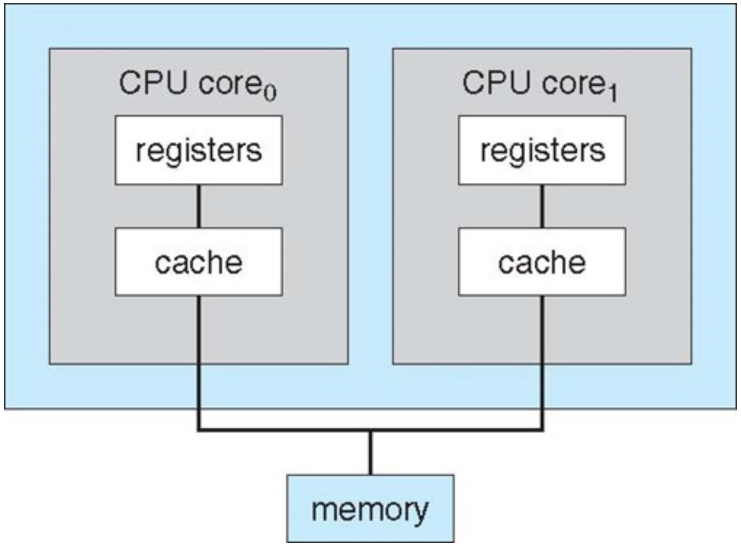
- Most systems use a single general-purpose processor
  - Most systems have special-purpose processors as well

- **Multiprocessors** systems growing in use and importance
  - Also known as **parallel systems**, **tightly-coupled systems**
  - Advantages include:
    1. Increased throughput
    2. Economy of scale
    3. Increased reliability – graceful degradation or fault tolerance
  - Two types:
    1. Asymmetric Multiprocessing – each processor is assigned a specific task.
    2. Symmetric Multiprocessing – each processor performs all tasks

Multiprocessing Architectures



Symmetric Multiprocessor:
loosely coupled, multiple chips



Dual-core Processors:
tightly coupled, single chip

Quiz

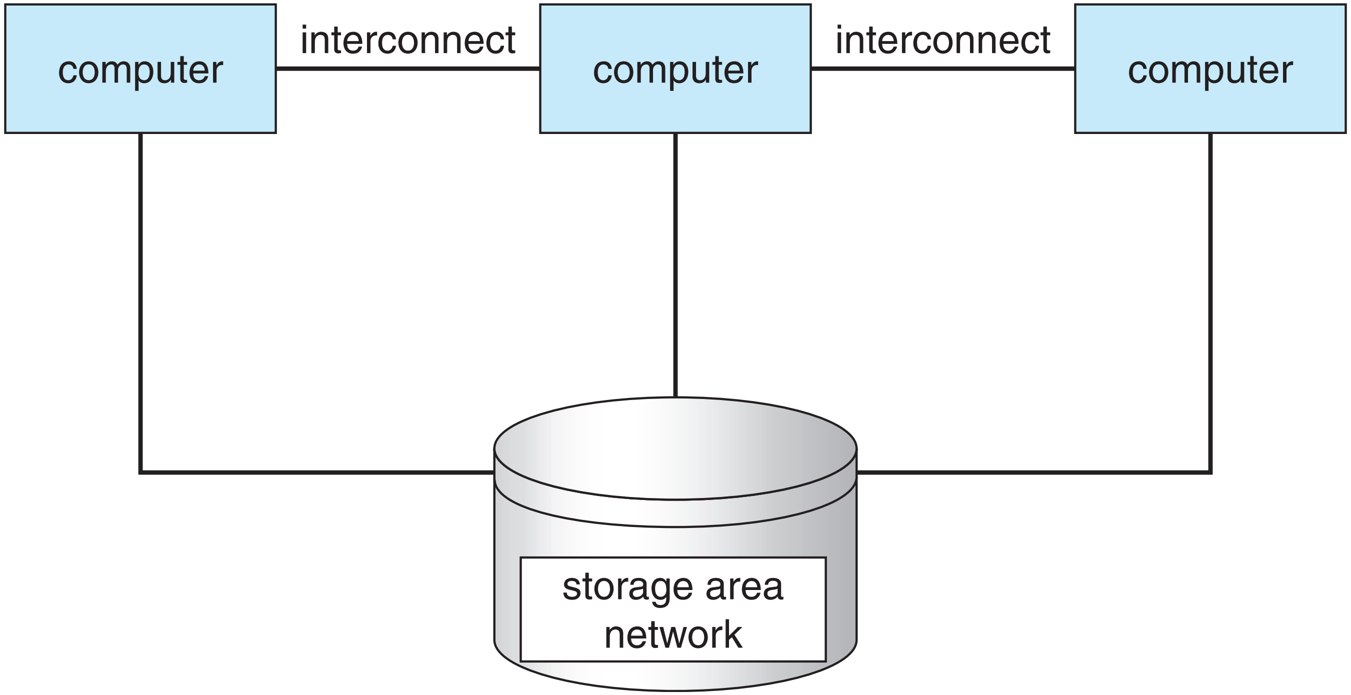
- Describe the differences between symmetric and asymmetric multiprocessing.

- What are three advantages and one disadvantage of multiprocessor systems?

Clustered Systems

- Like multiprocessor systems, but multiple systems working together
  - Usually sharing storage via a storage-area network (SAN)
  - Provides a **high-availability** service, i.e., service that will continue even if one or more systems in the cluster fail
    - **Asymmetric clustering** has one machine in hot-standby mode
    - **Symmetric clustering** has multiple nodes running applications, monitoring each other
  - Some clusters are for **high-performance computing (HPC)**
    - Applications must be written to use **parallelization**
  - Some have **distributed lock manager** (**DLM**) to avoid conflicting operations

Clustered Systems (Cont'd)



Quiz

- How do clustered systems differ from multiprocessor systems?

Types of OS (1)

- Batch operating system
  - The users of a batch operating system do not interact with the computer directly. Each user prepares his job on an off-line device like punch cards and submits it to the computer operator. To speed up processing, jobs with similar needs are batched together and run as a group.
  - The problems with Batch Systems are as follows −
    - Lack of interaction between the user and the job.
    - CPU is often idle, because the speed of the mechanical I/O devices is slower than the CPU.
    - Difficult to provide the desired priority.

Types of OS (2)

- Time-sharing operating systems
  - Time-sharing is a technique which enables many people, located at various terminals, to use a particular computer system at the same time. Time-sharing or multitasking is a logical extension of multiprogramming.
  - Advantages of Time-sharing operating systems are:
    - Provides the advantage of quick response.
    - Avoids duplication of software.
    - Reduces CPU idle time.
  - Disadvantages of Time-sharing operating systems are:
    - ➢ Problem of reliability.
    - ➢ Question of security and integrity of user programs and data.
    - ➢ Problem of data communication.

Types of OS (3)

- Distributed operating System
  - Distributed systems use multiple central processors to serve multiple real-time applications and multiple users. Data processing jobs are distributed among the processors accordingly.
  - The advantages of distributed systems are as follows −
    - With resource sharing facility, a user at one site may be able to use the resources available at another.
    - Speedup the exchange of data with one another via electronic mail.
    - If one site fails in a distributed system, the remaining sites can potentially continue operating.
    - Better service to the customers.
    - Reduction of the load on the host computer.
    - Reduction of delays in data processing

Types of OS (4)

- Network operating System
  - A Network Operating System runs on a server and provides the server the capability to manage data, users, groups, security, applications, and other networking functions. The primary purpose of the network operating system is to allow shared file and printer access among multiple computers in a network, typically a local area network (LAN), a private network or to other networks.
  - The advantages of network operating systems are:
    - Centralized servers are highly stable.
    - Security is server managed.
    - Upgrades to new technologies and hardware can be easily integrated into the system.
    - Remote access to servers is possible from different locations and types of systems.
  - The disadvantages of network operating systems are:
    - High cost of buying and running a server.
    - Dependency on a central location for most operations.
    - Regular maintenance and updates are required.

Types of OS (5)

- Real Time operating System
  - A real-time system is defined as a data processing system in which the time interval required to process and respond to inputs is so small that it controls the environment.
  - There are two types of real-time operating systems.
    - **Hard real-time systems** - guarantee that critical tasks complete on time. In hard real-time systems, secondary storage is limited or missing and the data is stored in ROM. In these systems, virtual memory is almost never found.
    - **Soft real-time systems** - are less restrictive. A critical real-time task gets priority over other tasks and retains the priority until it completes.

Outline

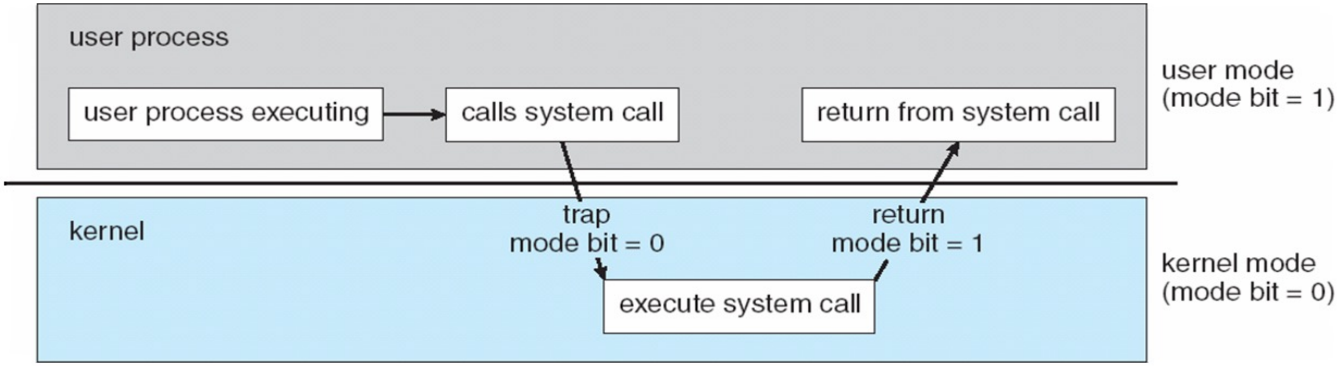
- 1.1 What Operating Systems Do
- 1.2 Computer-System Organization
- 1.3 Computer-System Architecture
- 1.4 Operating-System Operations
- 1.5 Resource Management

Operating System Operations

- **Interrupt driven** (hardware and software)
  - Hardware interrupt by one of the devices
  - Software interrupt (**exception** or **trap):**
    - Software error (e.g., division by zero)
    - Request for operating system service
    - Other process problems include infinite loop, processes modifying each other or the operating system

Operating-System Operations (Cont’d)

- **Dual-mode** operation allows OS to protect itself and other system components
  - **User mode** and **kernel mode**
  - **Mode bit** provided by hardware
    - Provides ability to distinguish when system is running user code or kernel code
    - Some instructions designated as **privileged**, only executable in kernel mode
    - System call changes mode to kernel, return from call resets it to user



Transition from User to Kernel Mode

- Timer to prevent infinite loop / process hogging resources
  - Timer is set to interrupt the computer after some time period
  - Keep a counter that is decremented by the physical clock.
  - Operating system set the counter (privileged instruction)
  - When counter zero generate an interrupt
  - Set up before scheduling process to regain control or terminate program that exceeds allotted time

Quiz

- Some computer systems do not provide a privileged mode of operation in hardware. Is it possible to construct a secure operating system for these computer systems? Give arguments both that it is and that it is not possible.
- Which of the following instructions should be privileged?
  a. Set value of timer.
  b. Read the clock.
  c. Clear memory.
  d. Issue a trap instruction.
  e. Turn off interrupts.
  f. Modify entries in device-status table.
  g. Switch from user to kernel mode.
  h. Access I/O device

Outline

- 1.1 What Operating Systems Do
- 1.2 Computer-System Organization
- 1.3 Computer-System Architecture
- 1.4 Operating-System Operations
- 1.5 Resource Management

Process Management

- The operating system manages many kinds of activities ranging from user programs to system programs like printer spooler, name servers, file server etc. Each of these activities is encapsulated in a process. A process includes the complete execution context (code, data, PC, registers, OS resources in use etc.).
- The five major activities of an operating system in regard to process management are:
  - Creation and deletion of user and system processes.
  - Suspension and resumption of processes.
  - A mechanism for process synchronization.
  - A mechanism for process communication.
  - A mechanism for deadlock handling.

Memory Management

- Primary-Memory or Main-Memory is a large array of words or bytes. Each word or byte has its own address. Main-memory provides storage that can be access directly by the CPU. A program to be executed, must be in the main memory.
- The major activities of an operating in regard to memory-management are:
  - Keep track of which part of memory are currently being used and by whom.
  - Decide which process are loaded into memory when memory space becomes available.
  - Allocate and deallocate memory space as needed.

File Management

- A file is a collected of related information defined by its creator. Computer can store files on the disk (secondary storage), which provide long term storage.. A file systems normally organized into directories to ease their use. These directories may contain files and other directions.
- The five main major activities of an operating system in regard to file management are:
  - The creation and deletion of files.
  - The creation and deletion of directions.
  - The support of primitives for manipulating files and directions.
  - The mapping of files onto secondary storage.
  - The back up of files on stable storage media.

Mass-Storage Management

- Systems have several levels of storage, including primary storage, secondary storage and cache storage. Instructions and data must be placed in primary storage or cache to be referenced by a running program. Secondary storage consists of tapes, disks, and other media.
- The three major activities of an operating system in regard to secondary storage management are:
  - Managing the free space available on the secondary-storage device.
  - Allocation of storage space when new files have to be written.
  - Scheduling the requests for memory access.

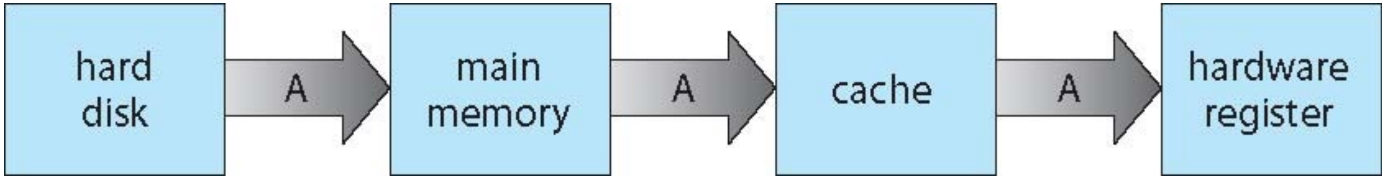
Performance of Various Levels of Storage

Level	1	2	3	4	5
Name	registers	cache	main memory	solid state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25 - 0.5	0.5 - 25	80 - 250	25,000 - 50,000	5,000,000
Bandwidth (MB/sec)	20,000 - 100,000	5,000 - 10,000	1,000 - 5,000	500	20 - 150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape

Movement between levels of storage hierarchy can be explicit or implicit

Migration of data “A” from Disk to Register

- Multitasking environments must be careful to use most recent value, no matter where it is stored in the storage hierarchy



- Multiprocessor environment must provide **cache coherency** in hardware such that all CPUs have the most recent value in their cache

- Distributed environment situation even more complex
  - Several copies of a datum can exist
  - Various solutions covered in Chapter 17

I/O Subsystem

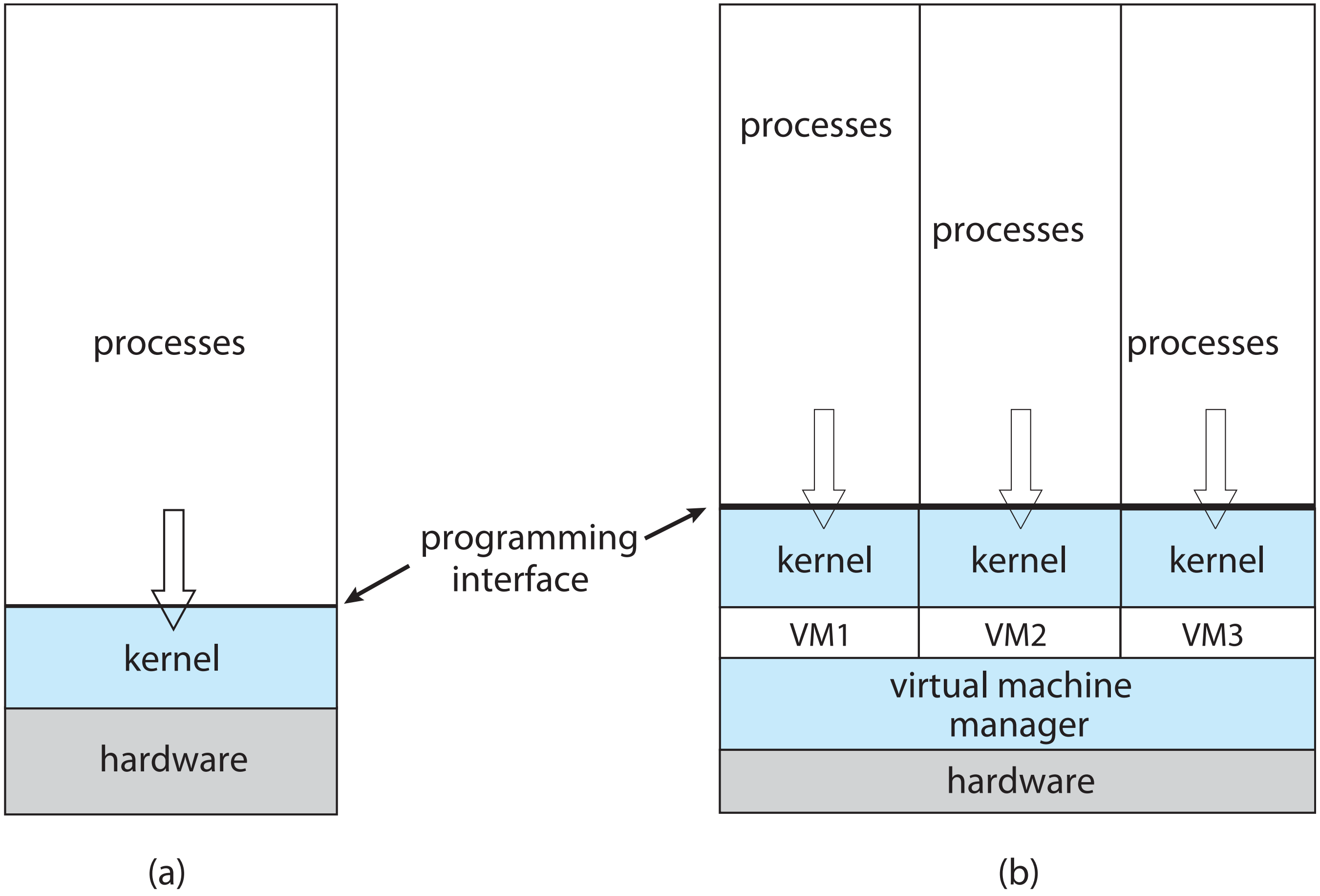
- One purpose of OS is to hide peculiarities of hardware devices from the user
- I/O subsystem responsible for
  - Memory management of I/O including buffering (storing data temporarily while it is being transferred), caching (storing parts of data in faster storage for performance), spooling (the overlapping of output of one job with input of other jobs)
  - General device-driver interface
  - Drivers for specific hardware devices

1.6 Security and Protection

- **Protection -** If a computer systems has multiple users and allows the concurrent execution of multiple processes, then the various processes must be protected from one another's activities. Protection refers to mechanism for controlling the access of programs, processes, or users to the resources defined by a computer systems.

- **Security** – defense of the system against internal and external attacks
  - Huge range, including denial-of-service, worms, viruses, identity theft, theft of service

1.7 Virtualization



1.8 Distributed Systems

- Distributed computing
  - Collection of separate, possibly heterogeneous, systems networked together
    - **Network** is a communications path, **TCP/IP** most common
      - **Local Area Network** (**LAN**)
      - **Wide Area Network** (**WAN**)
      - **Metropolitan Area Network** (**MAN**)
      - **Personal Area Network** (**PAN**)
  - **Network Operating System** provides features between systems across network
    - Communication scheme allows systems to exchange messages
    - Illusion of a single system

1.10 Computing Environments

- Traditional

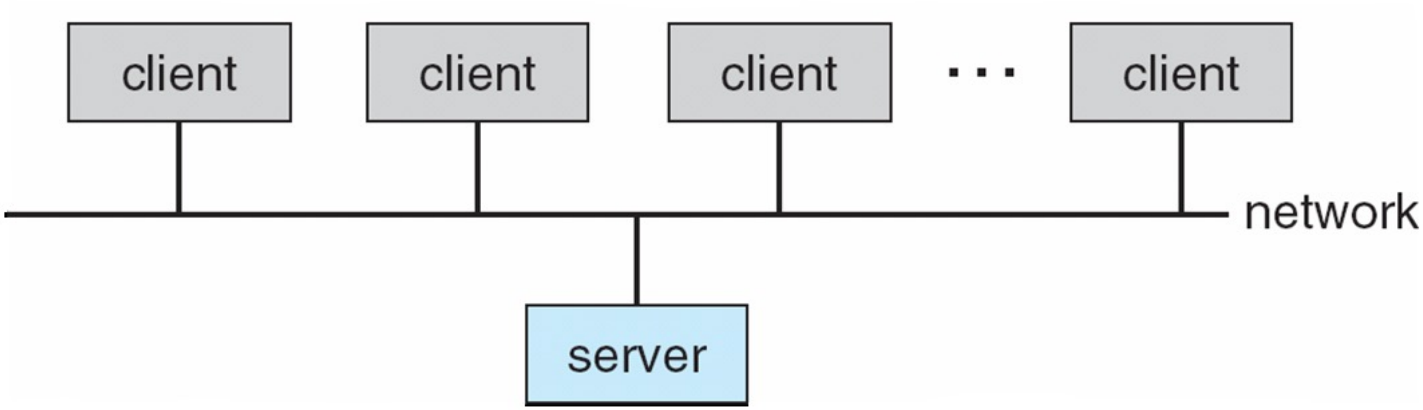
- Stand-alone general purpose machines
- But blurred as most systems interconnect with others (i.e., the Internet)
- **Portals** provide web access to internal systems
- **Network computers** (**thin clients**) are like Web terminals
- Mobile computers interconnect via **wireless networks**
- Networking becoming ubiquitous – even home systems use **firewalls** to protect home computers from Internet attacks

Computing Environments - Mobile

- Handheld smartphones, tablets, etc.
- What is the functional difference between them and a “traditional” laptop?
- Extra feature – more OS features
- Allows new types of apps like ***augmented reality***
- Use IEEE 802.11 wireless, or cellular data networks for connectivity
- Leaders are **Apple iOS** and **Google Android**

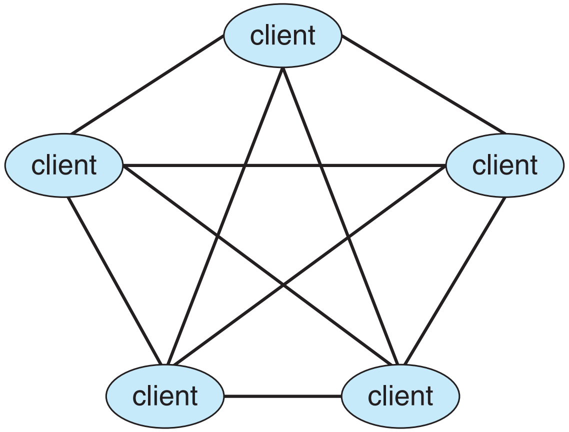
Computing Environments – Client-Server

- Client-Server Computing
  - Dumb terminals supplanted by smart PCs
  - Many systems now **servers**, responding to requests generated by **clients**
    - **Compute-server system** provides an interface to client to request services (i.e., database)
    - **File-server system** provides interface for clients to store and retrieve files



Computing Environments - Peer-to-Peer

- Another model of distributed system
- P2P does not distinguish clients and servers
  - Instead all nodes are considered peers
  - May each act as client, server or both
  - Node must join P2P network
    - Registers its service with central lookup service on network, or
    - Broadcast request for service and respond to requests for service via ***discovery protocol***
  - Examples include Napster and Gnutella*,* **Voice over IP** (**VoIP**) such as Skype



Computing Environments – Web-Based

- PCs most prevalent devices
- More devices becoming networked to allow web access
- New category of devices to manage web traffic among similar servers: **load balancers**
- Use of operating systems like Windows 95, client-side, have evolved into Linux and Windows XP, which can be clients and servers

Computing Environments - Virtualization

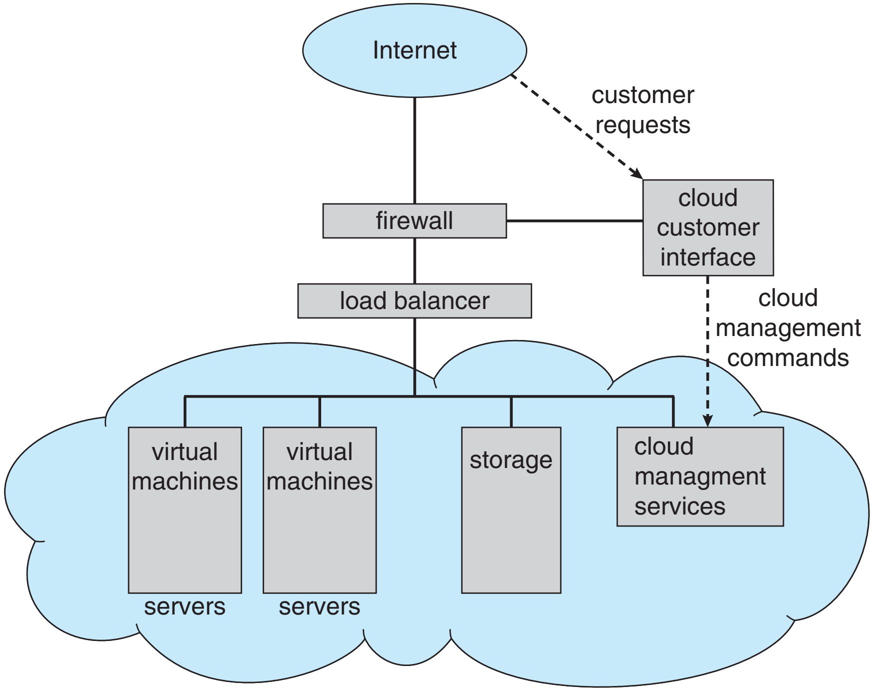
- Allows operating systems to run applications within other OSes
- **Virtualization** – OS natively compiled for CPU, running **guest** OSes also natively compiled
  - Consider VMware running WinXP guests, each running applications, all on native WinXP **host** OS
  - **VMM** (virtual machine Manager) provides virtualization services
  - Use cases involve laptops and desktops running multiple OSes for exploration or compatibility

Computing Environments – Cloud Computing

- Delivers computing, storage, even apps as a service across a network
- Logical extension of virtualization because it uses virtualization as the base for it functionality.
  - Amazon **EC2** has thousands of servers, millions of virtual machines, petabytes of storage available across the Internet, pay based on usage
- Many types
  - **Public cloud** – available via Internet to anyone willing to pay
  - **Private cloud** – run by a company for the company's own use
  - **Hybrid cloud** – includes both public and private cloud components
  - Software as a Service (**SaaS**) – one or more applications available via the Internet (i.e., word processor)
  - Platform as a Service (**PaaS**) – software stack ready for application use via the Internet (i.e., a database server)

Computing Environments – Cloud Computing

- Cloud computing environments composed of traditional OSes, plus VMMs, plus cloud management tools
  - Internet connectivity requires security like firewalls
  - Load balancers spread traffic across multiple applications



1.11 Open-Source Operating Systems

- Operating systems made available in source-code format rather than just binary **closed-source**

- Examples include **GNU/Linux** and **BSD UNIX** (including core of **Mac OS X**), and many more

- Advantages:
  - More secure
  - Community of interested programmers

Evolution of OSes

- Operating systems have evolved through a number of distinct phases or generations which corresponds roughly to the decades.

- **The 1940's - First Generations -** The earliest electronic digital computers had no operating systems..
- **The 1950's - Second Generation -** By the early 1950's, the routine had improved somewhat with the introduction of punch cards. The General Motors Research Laboratories implemented the first operating systems in early 1950's for their IBM 701.
- **The 1960's - Third Generation -** The systems of the 1960's were also batch processing systems, but they were able to take better advantage of the computer's resources by running several jobs at once. So operating systems designers developed the concept of multiprogramming in which several jobs are in main memory at once; a processor is switched from job to job as needed to keep several jobs advancing while keeping the peripheral devices in use.

Evolution of OSs

- **Fourth Generation -** With the development of LSI (Large Scale Integration) circuits, chips, operating system entered in the system entered in the personal computer and the workstation age. Microprocessor technology evolved to the point that it become possible to build desktop computers as powerful as the mainframes of the 1970s. Two operating systems have dominated the personal computer scene: MS-DOS, written by Microsoft and UNIX, which is dominant on the large personal computers using the Motorola 6899 CPU family.