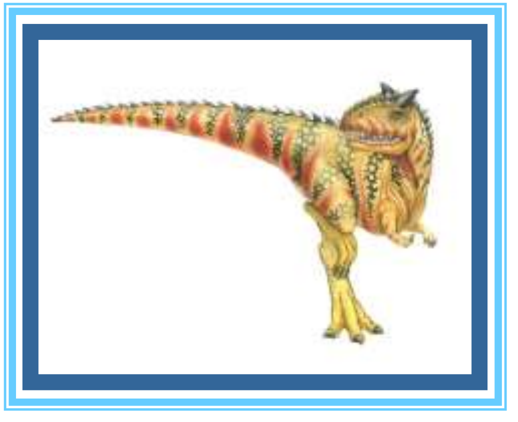
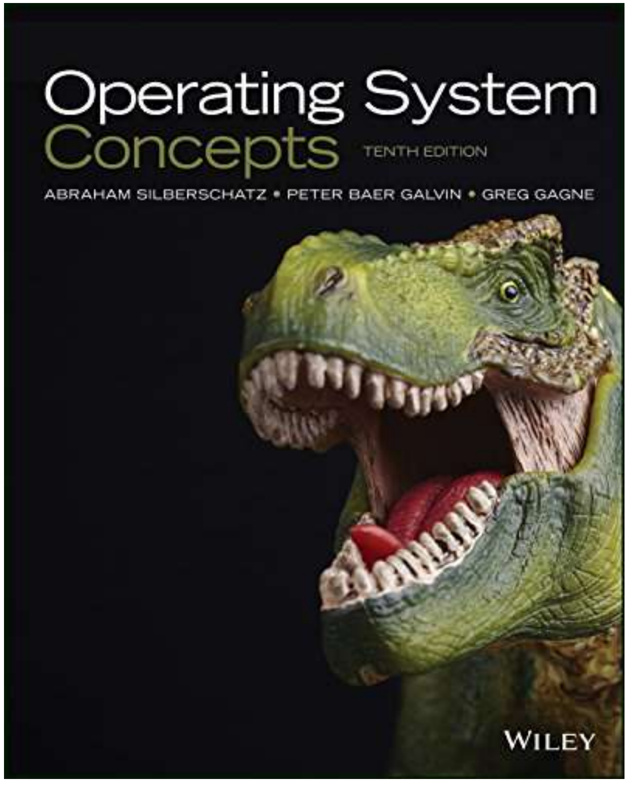
Operating Systems

Chapter 1: Introduction



Lecturer: Dalya Samer

Lectures Reference



Book by Avi Silberschatz, Greg Gagne, and Peter Baer Galvin

Course Objectives

- To describe the basic organization of computer systems.
- To describe the services an operating system provides to users, processes, and other systems.
- To discuss the various ways of structuring an operating system.
- To introduce the notion of a process and a thread.
- To introduce CPU scheduling, which is the basis for multiprogrammed operating systems.
- To develop a description of deadlocks.
- To provide a detailed description of various ways of organizing memory hardware.

Course Syllabus

- Introduction.
- Operating-System Structures.
- Processes.
- Threads.
- CPU Scheduling.
- Deadlocks.

Chapter 1: Introduction

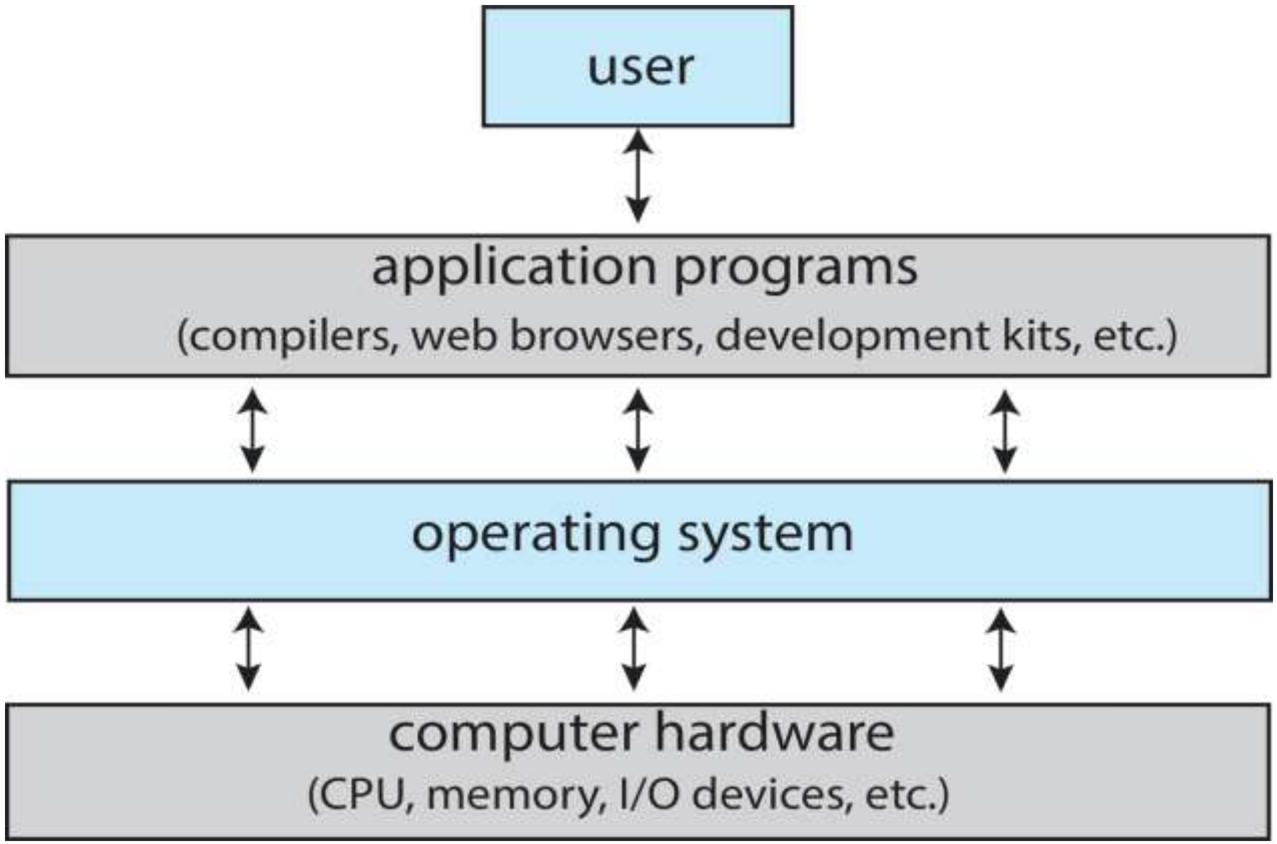
- Computer System Structure.
- What is an Operating System?
- What Operating Systems Do?
- Computer System Organization.
- Storage Structure.
- Multiprocessing Architecture.
- Operating-System Operations.
- Protection and Security.
- Computing Environments.

Computer System Structure

Computer system can be divided into four components:

- Hardware – provides basic computing resources
  - ➢ CPU, memory, I/O devices
- 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. Ex. Word processors, compilers, web browsers, database systems, video games.
- Users
  - ➢ People, machines, other computers

Abstract View of Components of Computer



What is an Operating System?

- An **operating system** is a program that manages a computer's hardware. It also provides a basis for application programs and acts as an intermediary between the computer user and the computer hardware.

- **Operating system 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.

What Operating Systems Do

User View

- Users want convenience, ease of use and good performance.
  - ➢ Don’t care about resource utilization.
- But shared computer such as mainframe or minicomputer must keep all users happy.

System View

- From the computer’s point of view, the operating system is the program most intimately involved with the hardware. In this context, we can view an operating system as a **resource allocator**.
- A computer system has many resources that may be required to solve a problem: CPU time, memory space, file-storage space, I/O devices, and so on. The operating system acts as the **manager** of these **resources**.

What Operating Systems Do

- OS is a **resource allocator**

  ➢ Manages all 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.

Operating System Definition

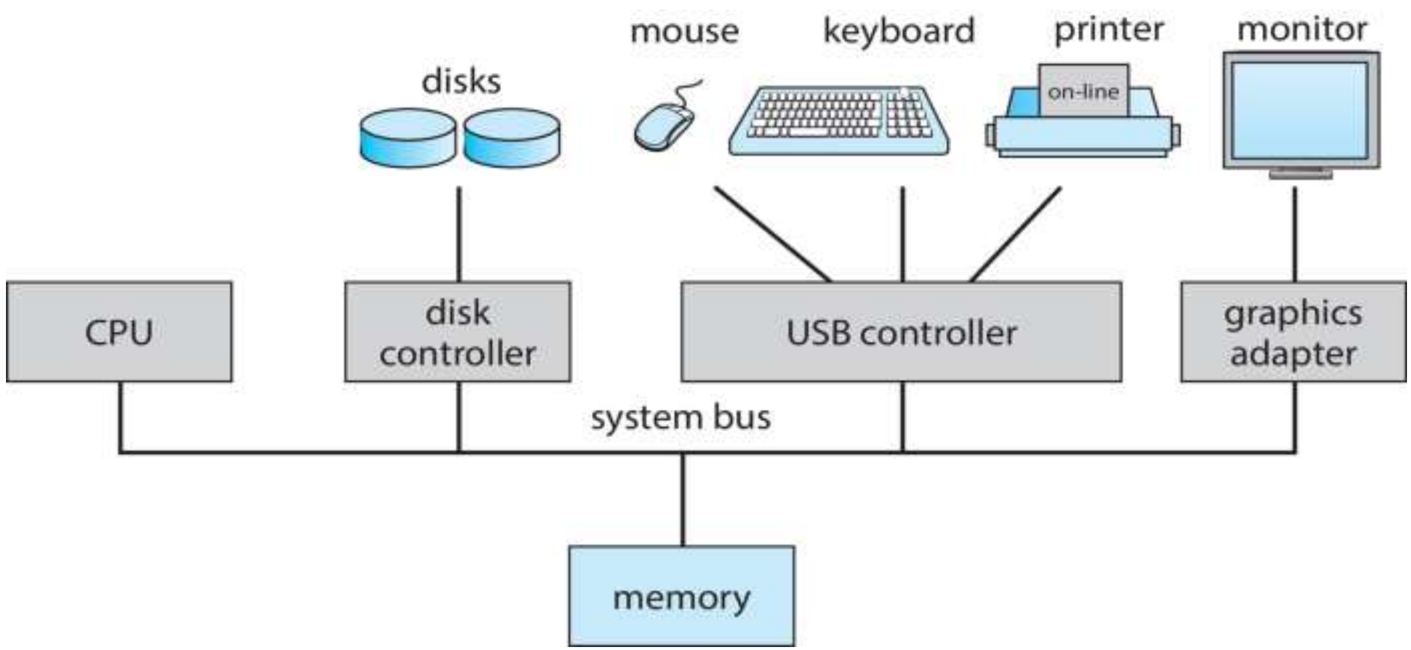
- A more common definition, and the one that we usually follow, is that the operating system is the one program running at all times on the computer—usually called the **kernel**.

- (Along with the kernel, there are two other types of programs:

  system programs, which are associated with the operating system but are not necessarily part of the kernel, and **application programs**, which include all programs not associated with the operation of the system.)

Computer System Organization

- A modern general-purpose computer system consists of one or more CPUs and a number of device controllers connected through a common bus that provides access to shared memory.



Computer-System Operation

Computer Startup (1/2)

- **bootstrap** program is loaded at power-up or reboot
  - ➢ Typically stored in ROM or electrically erasable programmable read-only memory (EPROM), generally known as **firmware**.
  - ➢ Initializes all aspects of system.
  - ➢ Loads operating system kernel and starts execution.

Computer Startup (2/2)

Common Functions of Interrupts

- **Interrupts (1/2)**

- The occurrence of an event is usually signaled by an **interrupt** from either the hardware or the software.
  - ➢ Hardware may trigger an interrupt at any time by sending a signal to the CPU, usually by way of the system bus.
  - ➢ Software may trigger an interrupt by executing a special operation called a **system call** (also called a **monitor call**).

- **Interrupts** are an important part of a computer architecture. Each computer design has its own interrupt mechanism, but several functions are common.

Interrupts (2/2)

- Interrupt transfers control to the interrupt service routine generally, through the **interrupt vector**, which contains the addresses of all the service routines.

- Interrupt architecture must save the address of the interrupted instruction.

- A **trap** or **exception** is a software-generated interrupt caused either by an error or a user request.

- An operating system is **interrupt driven.**

Storage Structure (1/4)

Review

The basic unit of computer storage is the **bit**. A bit can contain one of two values, 0 and 1. All other storage in a computer is based on collections of bits. Given enough bits, it is amazing how many things a computer can represent: numbers, letters, images, movies, sounds, documents, and programs, to name a few. A **byte** is 8 bits, and on most computers it is the smallest convenient chunk of storage. For example, most computers don't have an instruction to move a bit but do have one to move a byte. A less common term is **word**, which is a given computer architecture's native unit of data. A word is made up of one or more bytes. For example, a computer that has 64-bit registers and 64-bit memory addressing typically has 64-bit (8-byte) words. A computer executes many operations in its native word size rather than a byte at a time.

Computer storage, along with most computer throughput, is generally measured and manipulated in bytes and collections of bytes. A **kilobyte**, or **KB**, is 1,024 bytes; a **megabyte**, or **MB**, is $1{,}024^2$ bytes; a **gigabyte**, or **GB**, is $1{,}024^3$ bytes; a **terabyte**, or **TB**, is $1{,}024^4$ bytes; and a **petabyte**, or **PB**, is $1{,}024^5$ bytes. Computer manufacturers often round off these numbers and say that a megabyte is 1 million bytes and a gigabyte is 1 billion bytes. Networking measurements are an exception to this general rule; they are given in bits (because networks move data a bit at a time).

Storage Structure (2/4)

- 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

Storage Structure (3/4)

- **Hard Disk Drives** (**HDD**) – 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

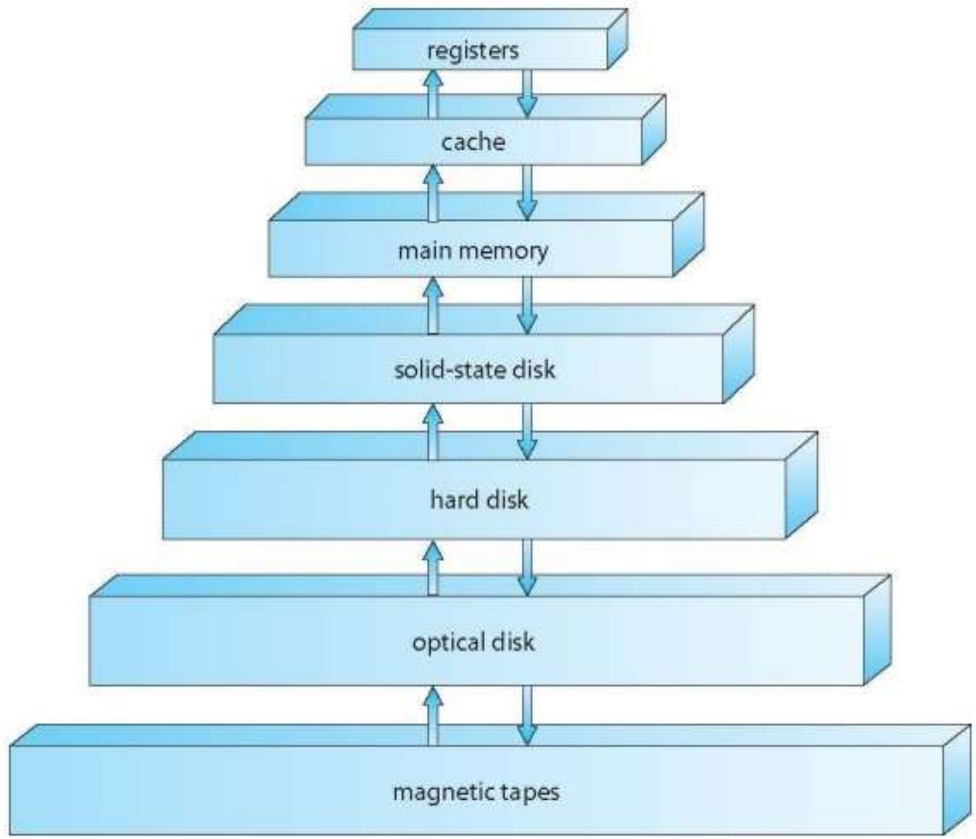
Storage Structure (4/4)

- Storage systems organized in hierarchy

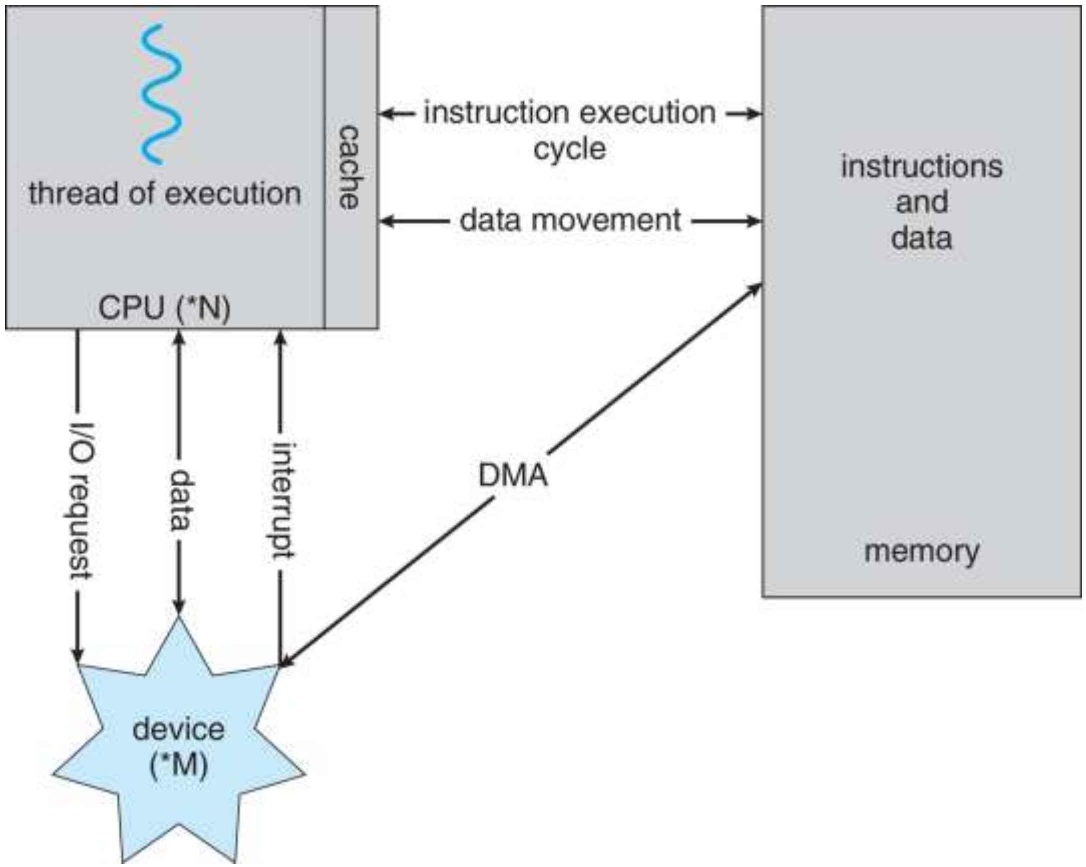
➢ Speed

➢ Cost

➢ Volatility



How a Modern Computer Works



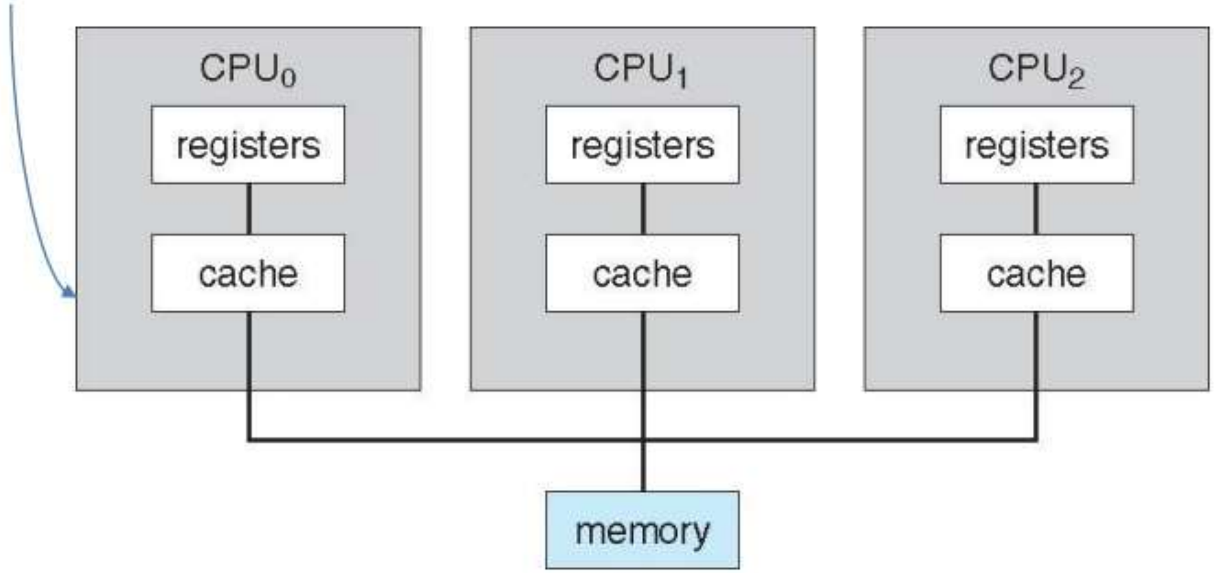
Multiprocessing Architecture(1/2)

- **Multiprocessors** systems growing in use and importance
  - Also known as **parallel systems.**
  - **Advantages include:**
    1. **Increased throughput** by increasing the number of processors, we expect to get more work done in less time.
    2. **Economy of scale** Multiprocessor systems can cost less than equivalent multiple single-processor systems, because they can share peripherals, mass storage, and power supplies.
    3. **Increased reliability** – graceful degradation or fault tolerance
  - **Two types:**
    1. **Asymmetric Multiprocessing** – each processor is assigned a specie task.
    2. **Symmetric Multiprocessing** – each processor performs all tasks

Multiprocessing Architecture(2/2)

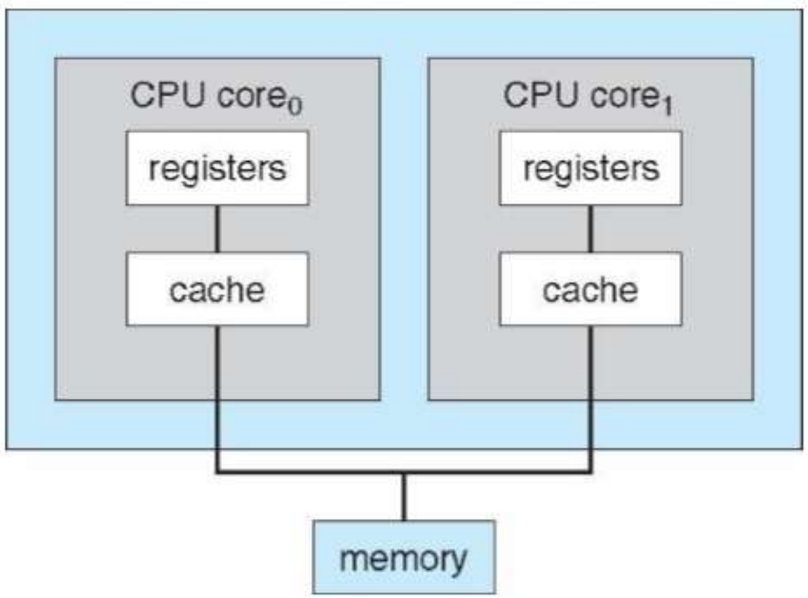
- **Multiprocessors** systems growing in use and importance
  - Two types:

1. **Asymmetric Multiprocessing** – each processor is assigned a specie task.
2. **Symmetric Multiprocessing** – each processor performs all tasks.



A Dual-Core Design

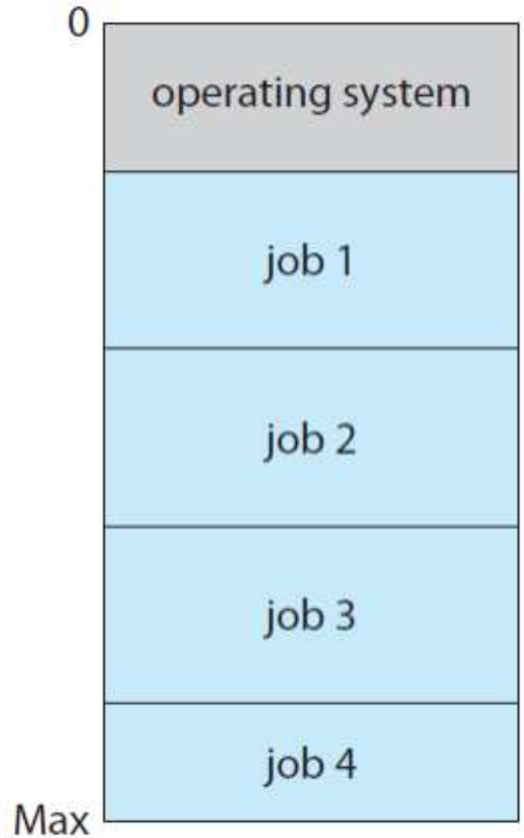
- **Multi-chip and multicore**



Operating System Structure (1/3)

- **Multiprogramming** (**Batch** system) needed for efficiency.
- Single user cannot always keep CPU and I/O devices busy
- Multiprogramming organizes jobs (code and data) so CPU always has one to execute
- A subset of total jobs in system is kept in memory
- One job selected and run via **job scheduling**
- When job has to wait (for I/O for example), OS switches to another job

Operating System Structure (2/3)



Operating System Structure (3/3)

- **Multitasking (Timesharing)** is a logical extension of Batch systems– the CPU switches jobs so frequently that users can interact with each job while it is running, creating **interactive** computing.

  - **Response time** should be < 1 second
  - Each user has at least one program executing in memory □ **process**
  - If several jobs ready to run at the same time □ **CPU scheduling**
  - If processes don't fit in memory, **swapping** moves them in and out to run
  - **Virtual memory** allows execution of processes not completely in memory

Operating-System Operations(1/3)

- **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(2/3)

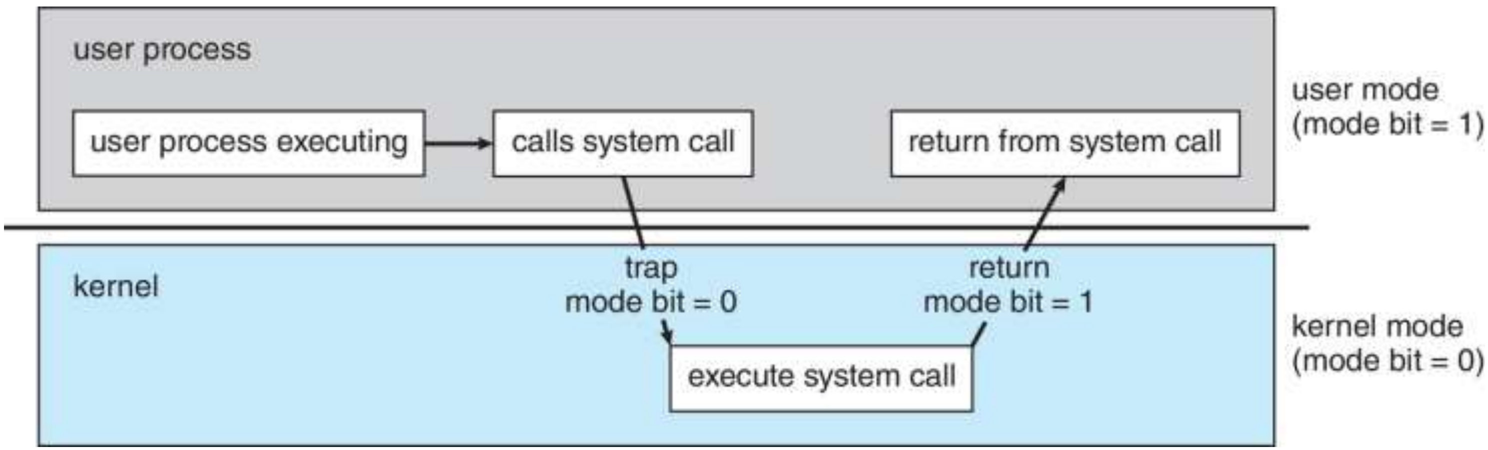
- **Dual-mode** operation allows OS to protect itself and other system components
  - **User mode** and **kernel mode** (also called supervisor mode, system mode, or privileged mode).

- A bit, called the mode bit, is added to the hardware of the computer to indicate the current mode: kernel (0) or user (1).

  - 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.

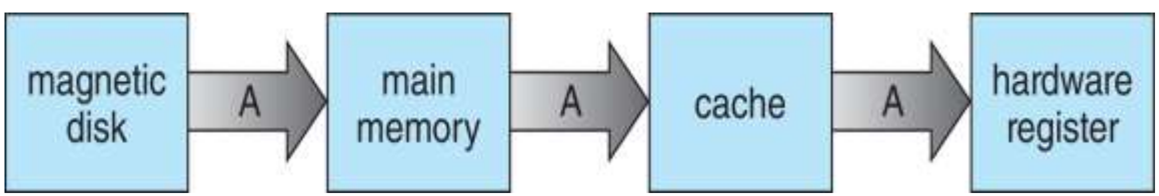
Operating-System Operations(3/3)

- **Transition from User to Kernel Mode**



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



Protection and Security

- **Protection** – any mechanism for controlling access of processes or users to resources defined by the OS
- **Security** – defense of the system against internal and external attacks
  - Huge range, including denial-of-service, worms, viruses, etc.

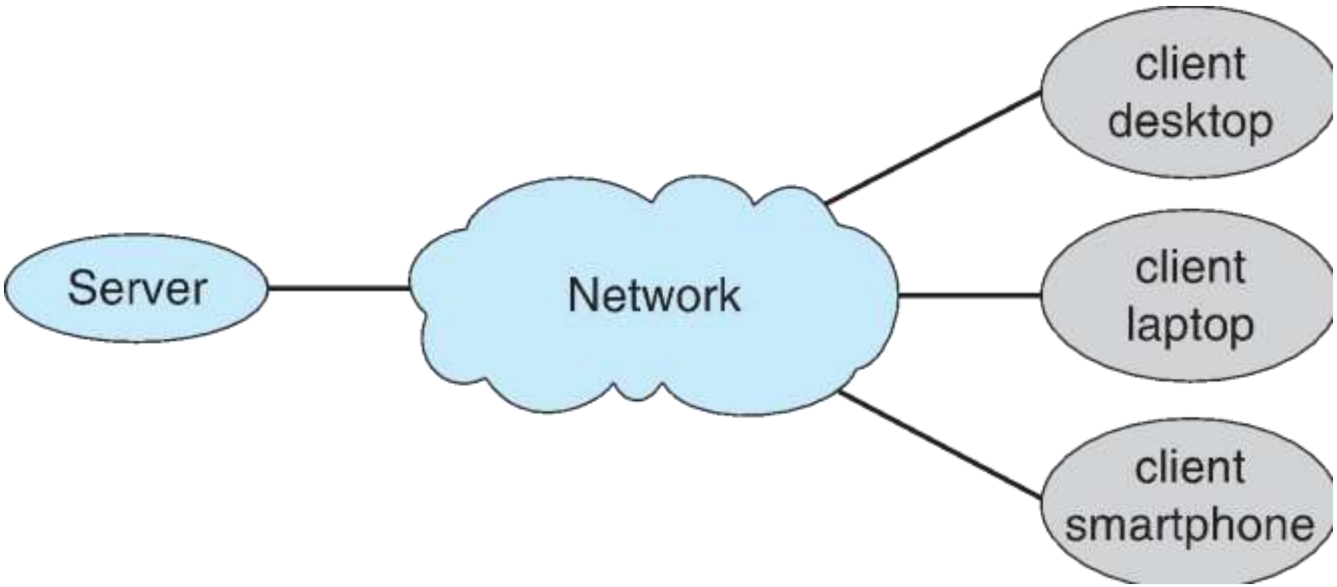
Computing Environments(1/9)

- The current trend is toward providing more ways to access these computing environments.
- Companies establish **portals,** which provide Web accessibility to their internal servers. **Portals** provide web access to internal systems
- **Network computers** (or **thin clients**) which are essentially terminals that understand web-based computing — are used in place of traditional workstations where more security or easier maintenance is desired.
- Mobile computers interconnect via **wireless networks**

Computing Environments(2/9)

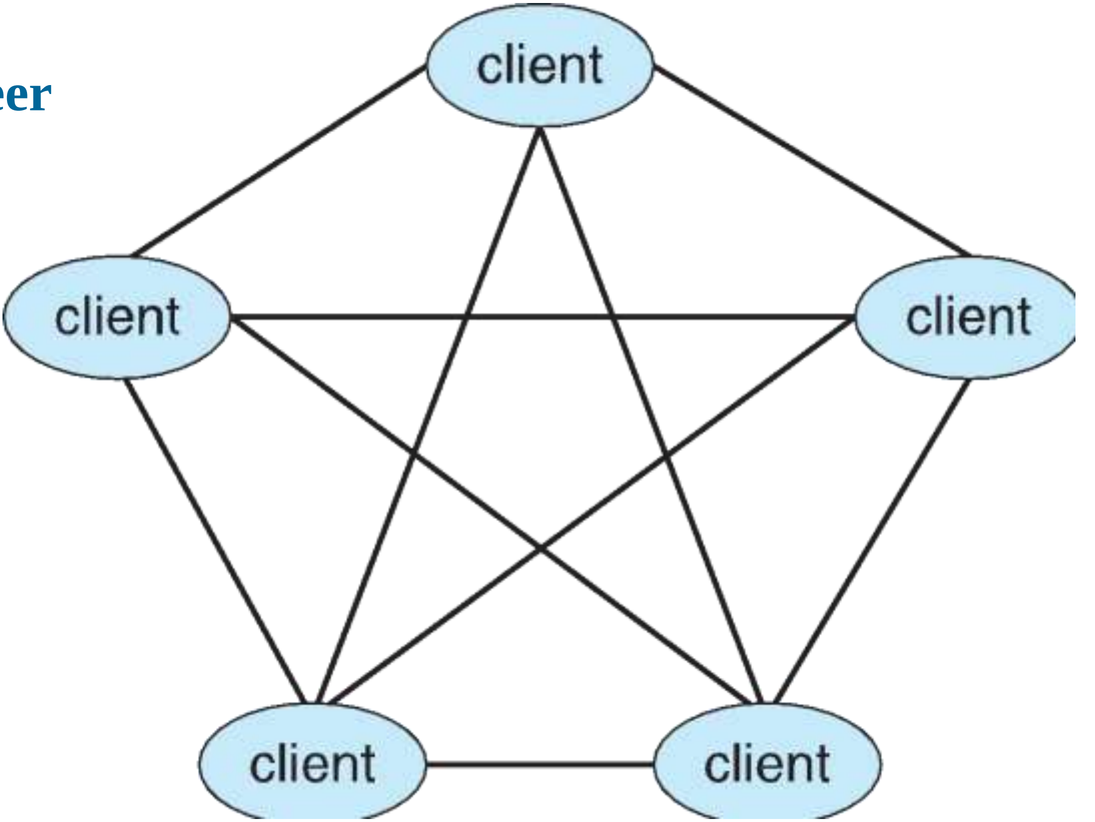
- **Distributed computing**: Collection of separate, possibly heterogeneous, systems networked together
  - **Network** is a communications path, **TCP/IP** most common
    - **Local Area Network** (**LAN**)
    - **Wide Area Network** (**WAN**)
    - **Metropolitan Area Network** (**MAN**)
    - **Personal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

Computing Environments(3/9)



Computing Environments(4/9)

- **Peer-to-Peer**



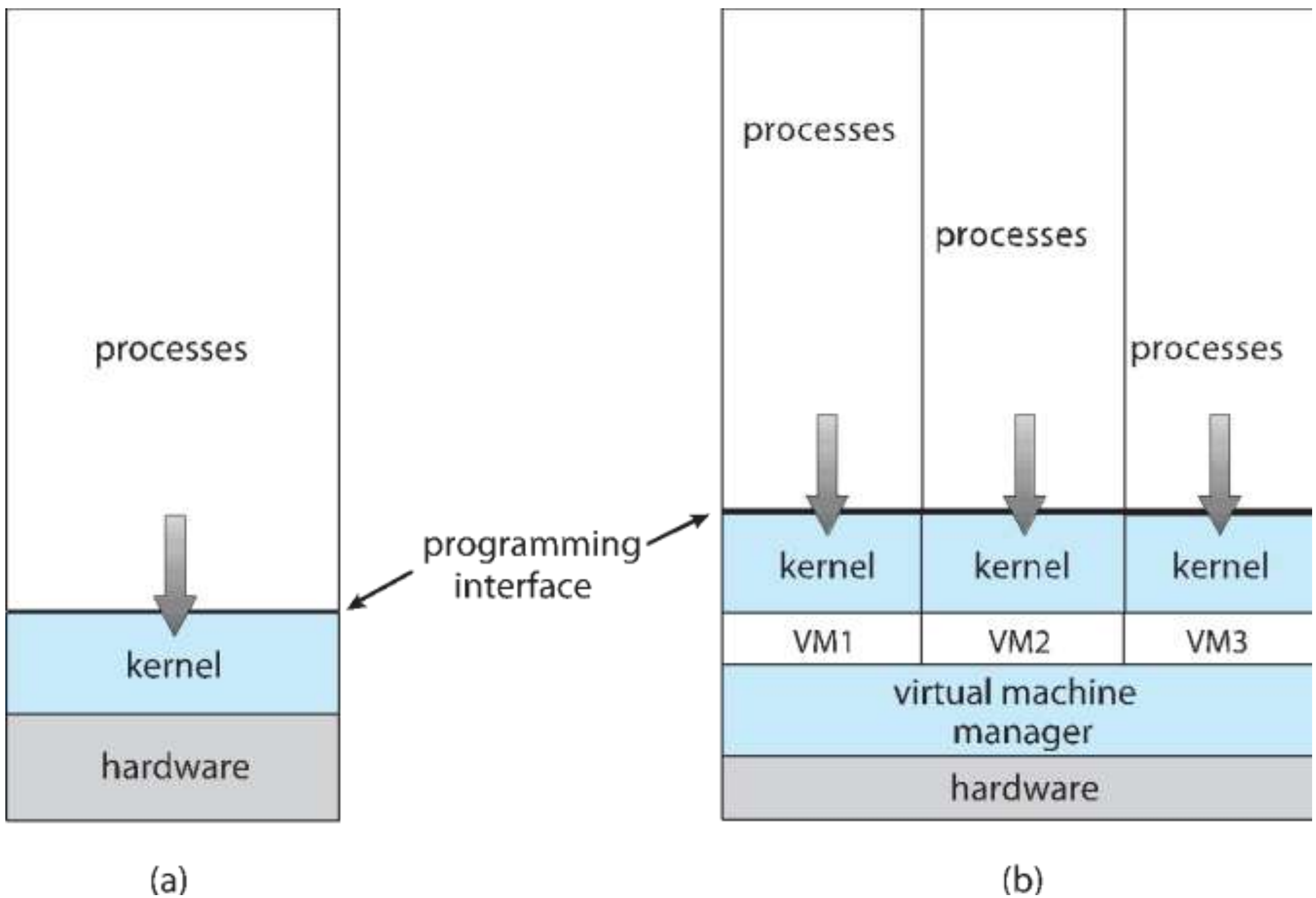
Computing Environments(5/9)

- **Virtualizations** a technology that allows operating systems to run applications within other Operating systems
  - Vast and growing industry

- 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.

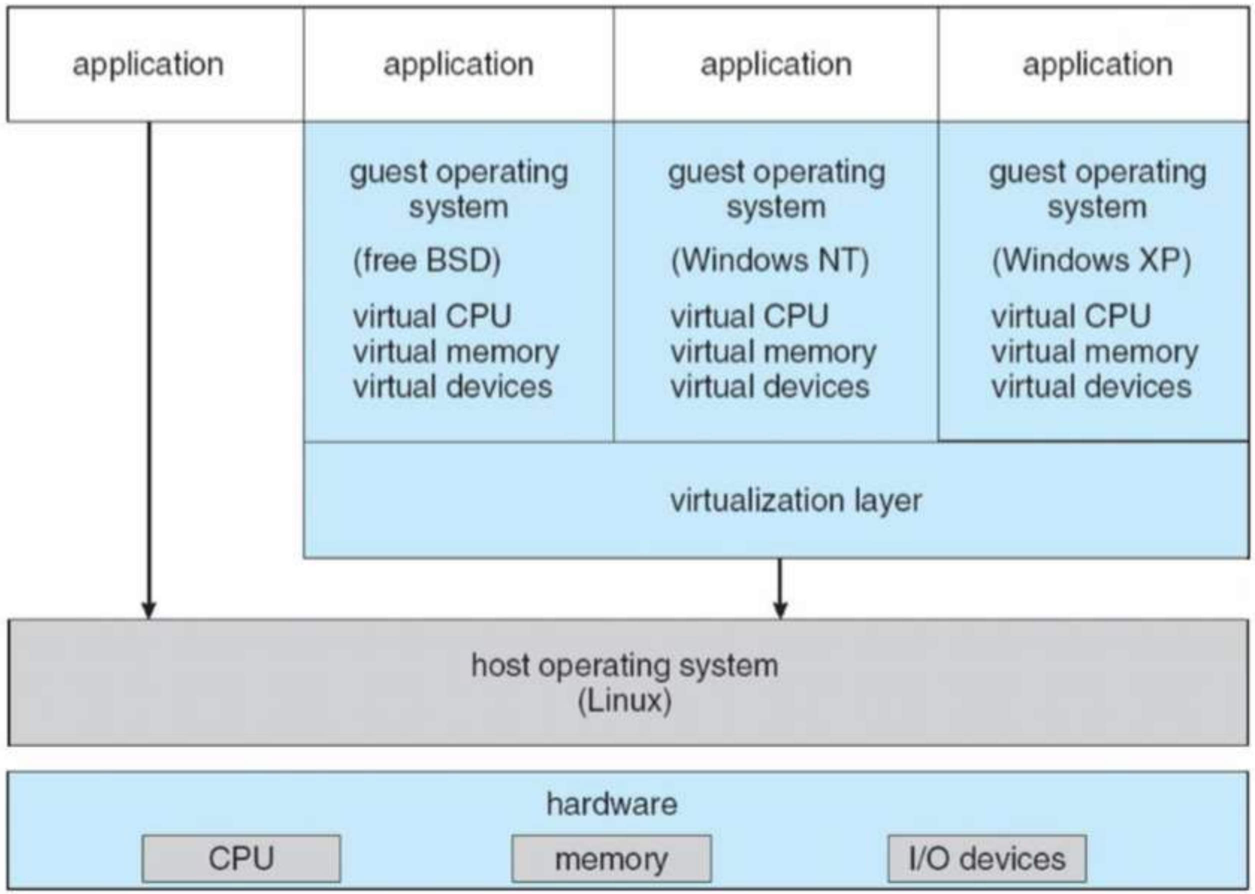
  - **VMM** (virtual machine Manager) provides virtualization services

Computing Environments(6/9)



Computing Environments(7/9)

- **VMware Architecture**



Computing Environments(8/9)

- **Cloud Computing-** Delivers computing, storage, even apps as a service across a network.

- **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.

Computing Environments(9/9)

- **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).

- **Infrastructure as a Service (IaaS)**

➢ Servers or storage available over Internet (i.e., storage available for backup use).

End of Chapter 1