CS-3103 : Operating Systems : Sec-A (NB) : Introduction

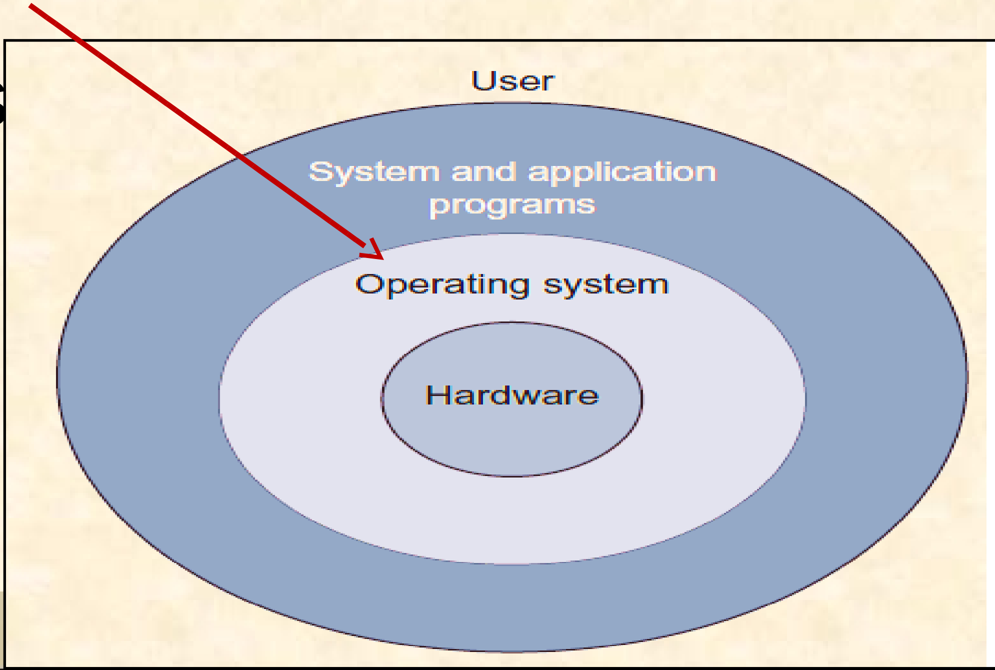
Chapter 1: Introduction

- What is an Operating System?
- Computer-System Organization
- Operating-System Structure
- Operating-System Operations
- Process Management
- Memory Management
- Storage Management
- Protection and Security

What is an OS with us every day (may be 24/7) ?

- When a computer is turned on, the **operating system (OS)** is loaded into its main memory and it takes control of the system.
- It provides an interface to the user for interaction.
- ***Set of programs that acts as an interface between the user and the machine.***
- ***It manages and allocates the resources available in the computer system, thus, OS can be considered as a resource manager.***
- Examples: Windows, Mac OS

Unix,Linux, etc.

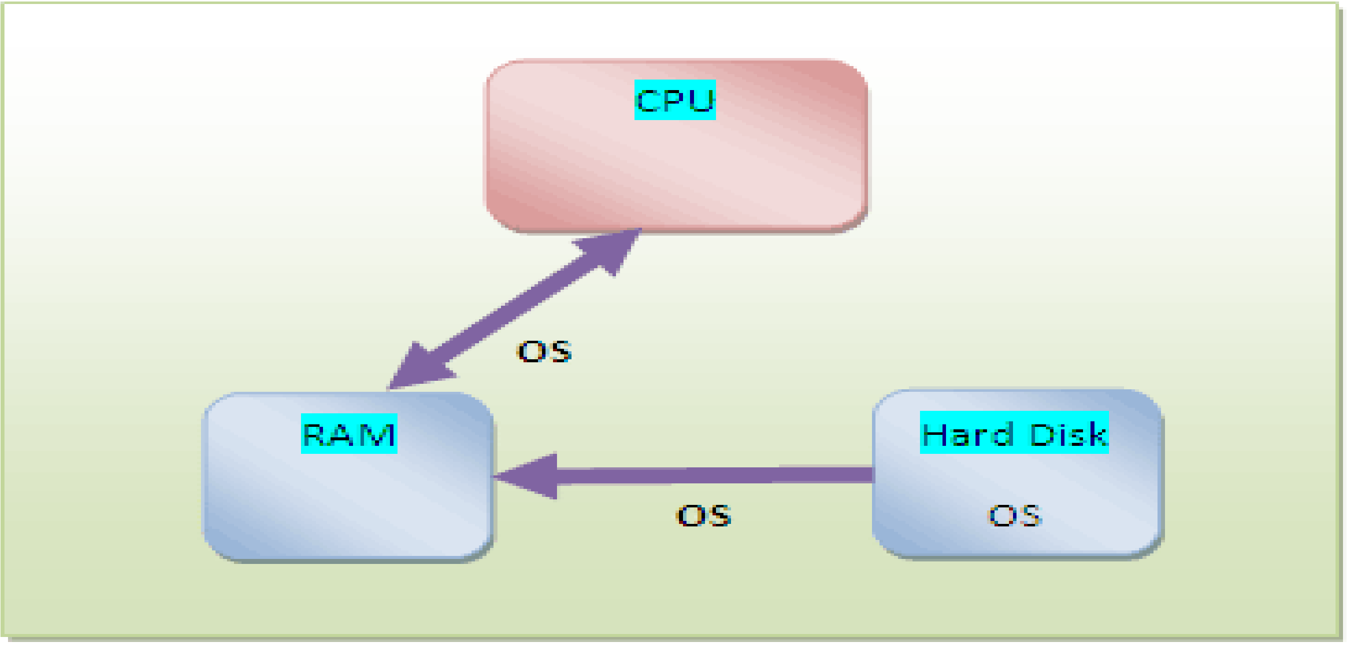


Computer System Structure

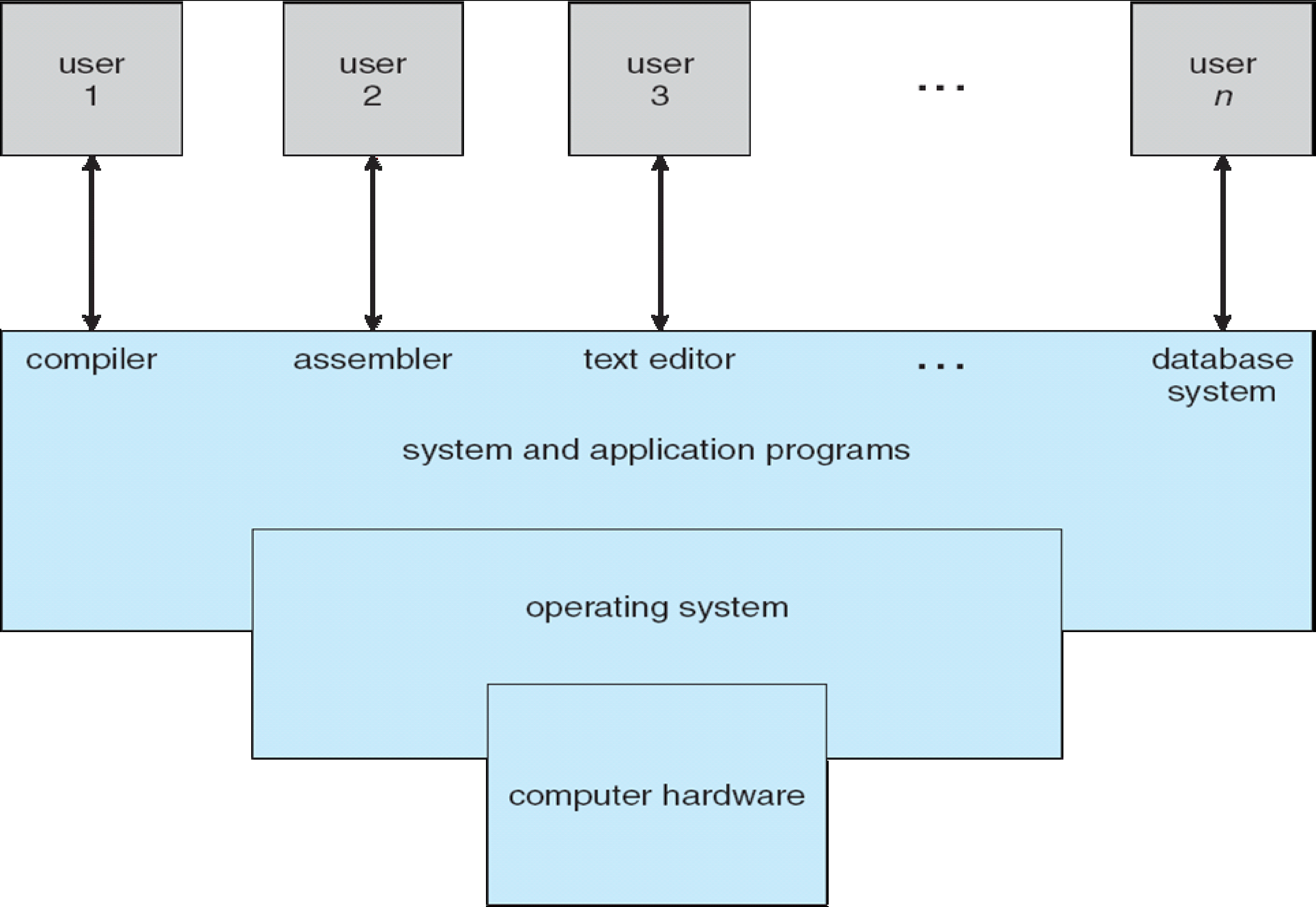
- Computer system can be divided into four components:
  - **Hardware** – provides basic computing resources
    - CPU, memory, I/O devices
  - **Operating system**
    - Controls and coordinates use of hardware among various applications and users
  - **Application programs** – define the ways in which the system resources are used to solve the computing problems of the users
    - Word processors, compilers, web browsers, database systems, video games
  - **Users**
    - People, machines, other computers

Objectives of an Operating System?

- **User-friendly**
- **Interprets commands**
- **Creates an environment for other programs to execute on the CPU**
- **Allocates resources to the executing program as and when required**
- **Harnesses the full power of the machine**



Four Components of a Computer System



User View

The user's view of the computer varies according to the interface being used

- **Single user computers** (e.g., PC, workstations). Such systems are designed for one user to monopolize its resources. The goal is to maximize the work (or play) that the user is performing. the operating system is designed mostly for **ease of use** and **good performance**.
- **Multi user computers** (e.g., mainframes, computing servers). These users share resources and may exchange information. The operating system in such cases is designed to maximize resource utilization -- to assure that all available CPU time, memory, and I/O are used efficiently and that no individual users takes more than their air share.

User View (Cont.)

- **Handheld computers** (e.g., smartphones and tablets). The user interface for mobile computers generally features a **touch screen.** The systems are resource poor, optimized for usability and battery life.
- **Embedded computers** (e.g., computers in home devices and automobiles) The user interface may have numeric keypads and may turn indicator lights on or off to show status. The operating systems are designed primarily to run without user intervention.

System View

From the computer's point of view, the operating system is the program most intimately involved with the hardware. There are two different views:

- The operating system is a **resource allocator**
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use
- The operating systems is a **control program**
  - Controls execution of programs to prevent errors and improper use of the computer

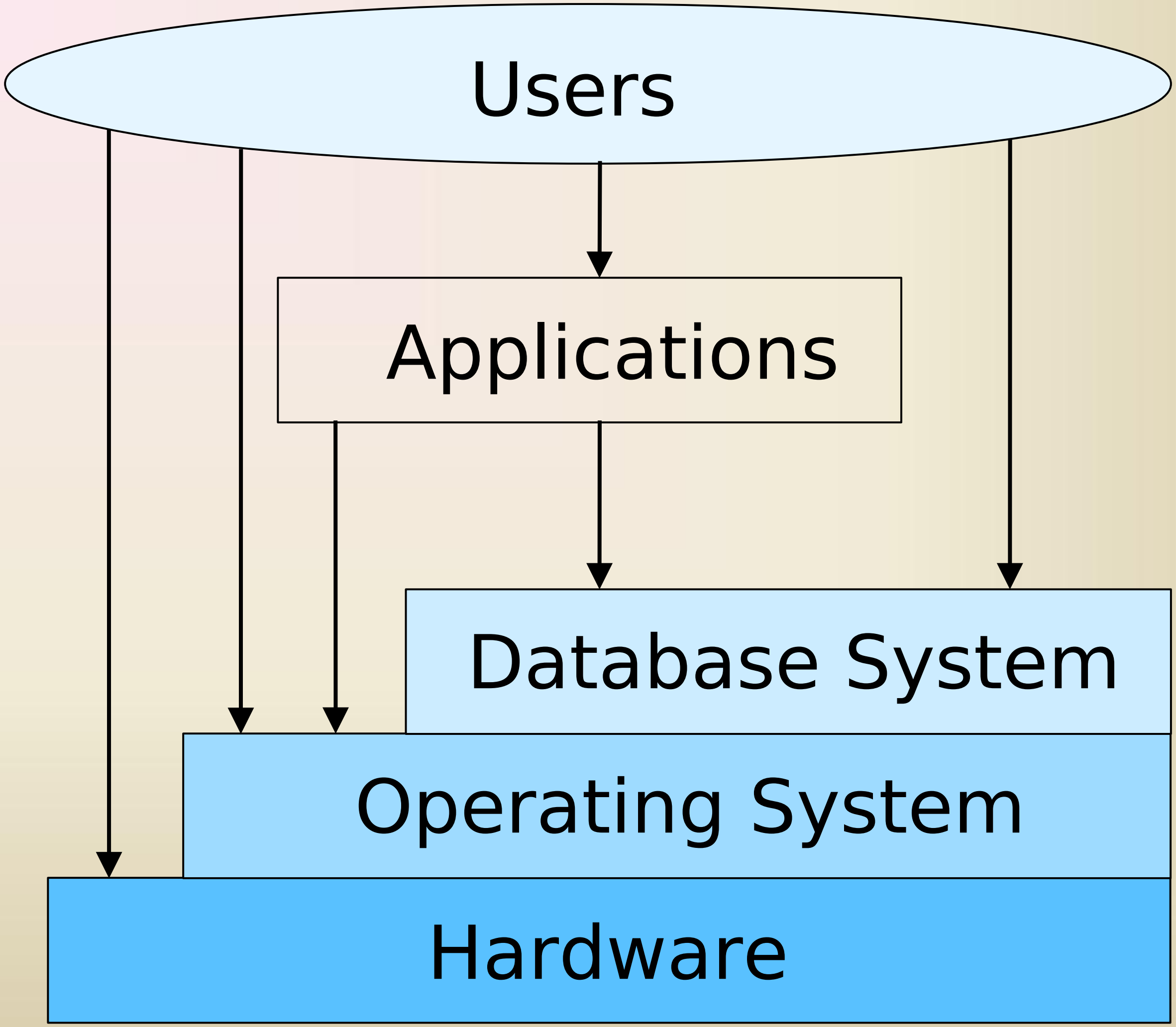
Let’s learn some more keywords regarding Operating Systems

- 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.
  - Application programs, which include all programs not associated with the operation of the system.

Let’s learn some more keywords regarding Operating Systems

- The emergence of mobile devices, have resulted in an increase in the number of features that constituting the operating system.
- **Mobile operating systems** often include not only a core kernel but also **middleware** -- a set of software frameworks that provide additional services to application developers.
- For example, each of the two most prominent mobile operating systems -- **Apple's iOS** and **Google's Android** -- feature a core kernel along with middleware that supports databases, multimedia, and graphics (to name only a few).

Evolution of Computer Systems



Users

Applications

Database System

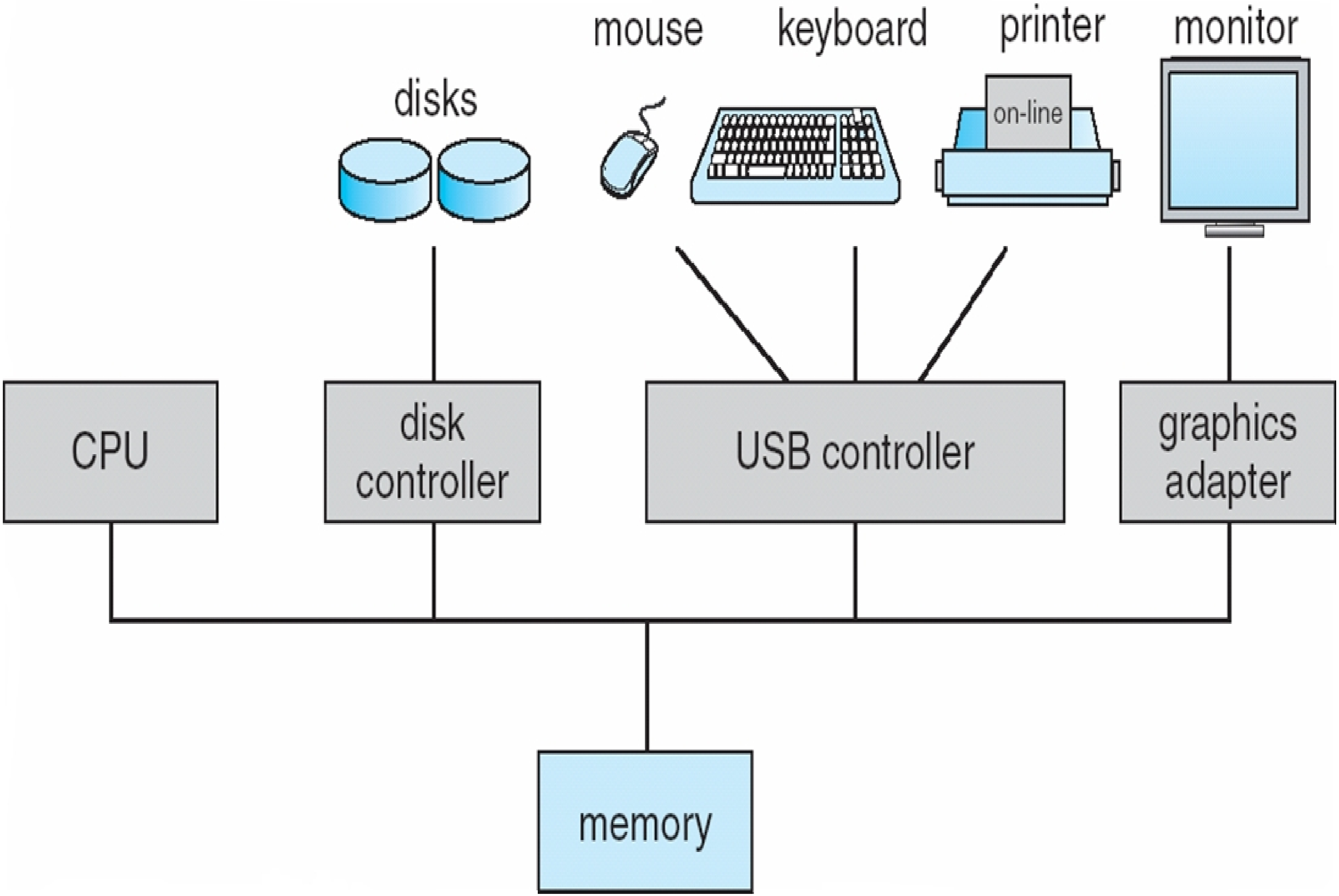
Operating System

Hardware

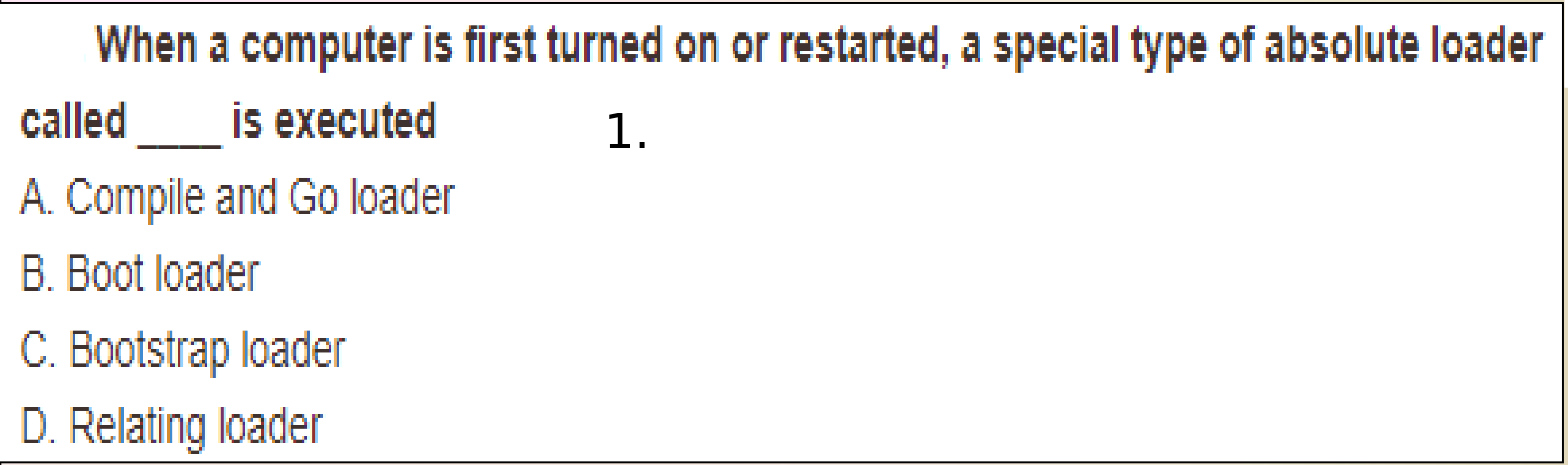
Computer-System Organization

- 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.
- Each device controller is in charge of a specific type of device (for example, disk drives, audio devices, or video displays). Each device controller has a local buffer.
- CPU moves data from/to main memory to/from local buffers.
- The CPU and the device controllers can execute in parallel, competing for memory cycles. To ensure orderly access to the shared memory, a memory controller synchronizes access to the memory.

Modern Computer System



Let’s relax a bit….



When a computer is first turned on or restarted, a special type of absolute loader called ____ is executed

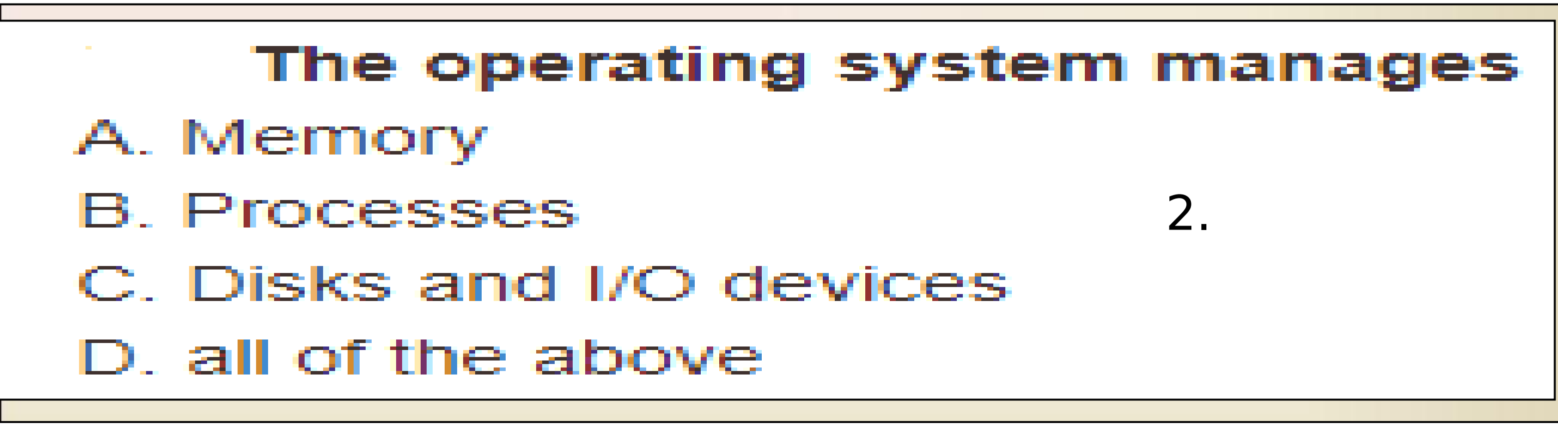
A. Compile and Go loader

B. Boot loader

C. Bootstrap loader

D. Relating loader

1.



The operating system manages

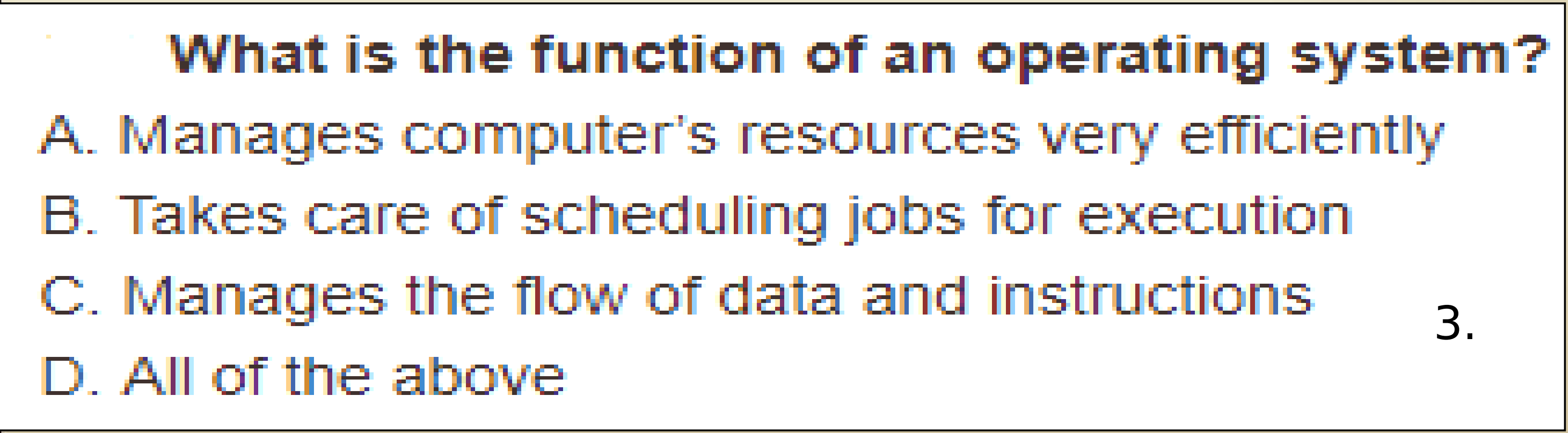
A. Memory

B. Processes

C. Disks and I/O devices

D. all of the above

2.



What is the function of an operating system?

A. Manages computer’s resources very efficiently

B. Takes care of scheduling jobs for execution

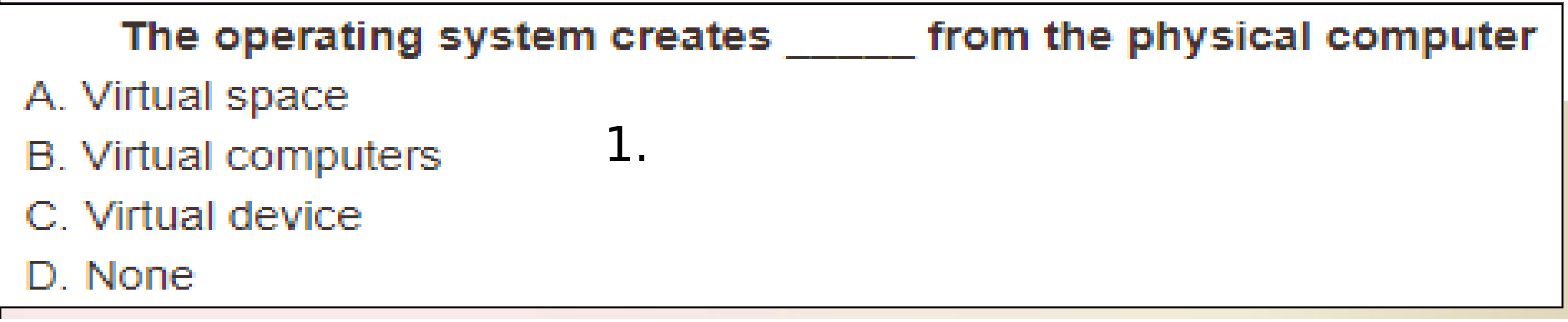
C. Manages the flow of data and instructions

D. All of the above

3.

1. c),
2. d),
3. d)

Let’s relax a bit….



The operating system creates ______ from the physical computer

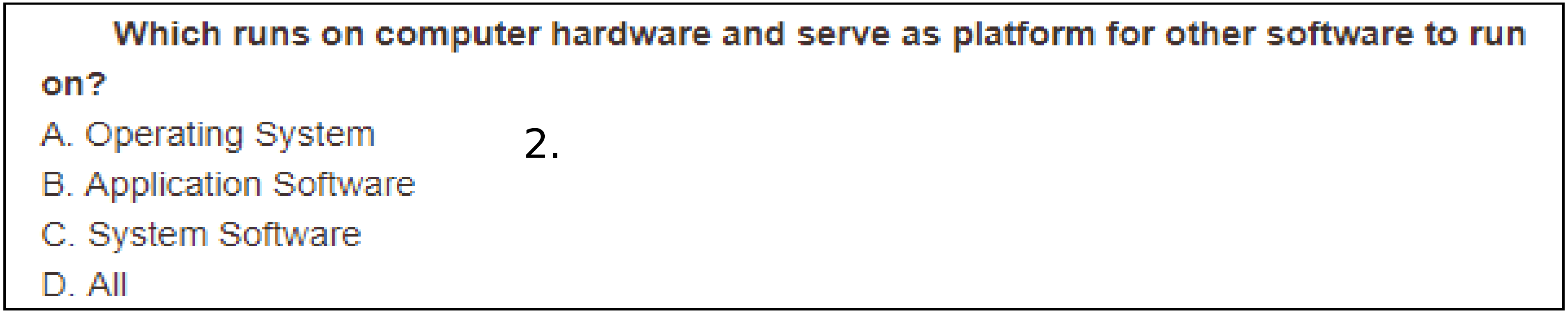
A. Virtual space

B. Virtual computers

C. Virtual device

D. None

1.



Which runs on computer hardware and serve as platform for other software to run on?

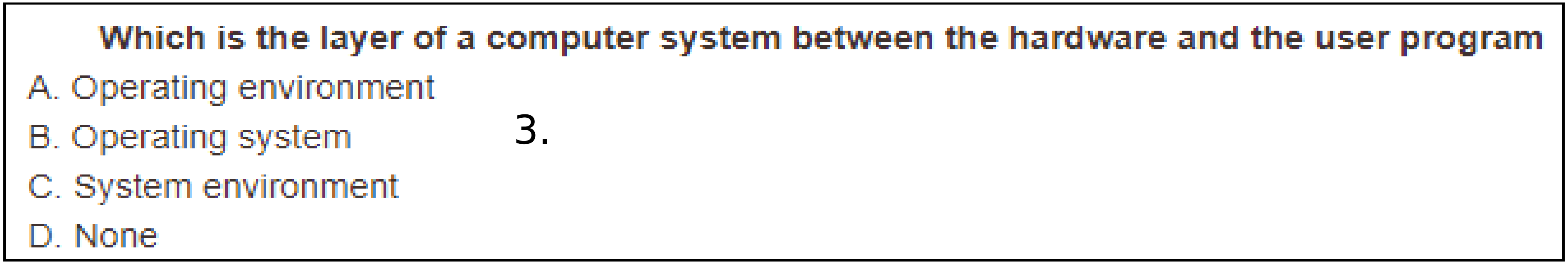
A. Operating System

B. Application Software

C. System Software

D. All

2.



Which is the layer of a computer system between the hardware and the user program

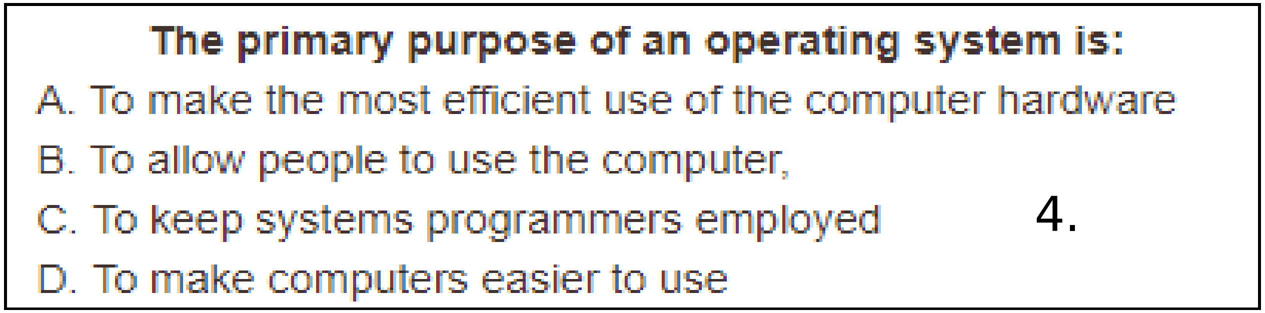
A. Operating environment

B. Operating system

C. System environment

D. None

3.



The primary purpose of an operating system is:

A. To make the most efficient use of the computer hardware

B. To allow people to use the computer,

C. To keep systems programmers employed

D. To make computers easier to use

4.

1. a), 2. a), 3. b), 4. a)

Computer Startup

- **Bootstrap program** is loaded at power-up or reboot
  - Typically stored in ROM or EPROM, generally known as **firmware**
  - Initializes all aspects of system
  - Loads operating system kernel and starts execution

Computer-System Operation

- Once the kernel is loaded and executing, it can start providing services to the system and its users.
- Some services are provided outside of the kernel, by system programs that are loaded into memory at boot time to become **system processes**, or **system daemons** that run the entire time the kernel is running.
- **On UNIX, the first system process is init** and it starts many other daemons. Once this phase is complete, the system is fully booted, and the system waits for some event to occur.
- The occurrence of an event is usually signaled by an **interrupt.**

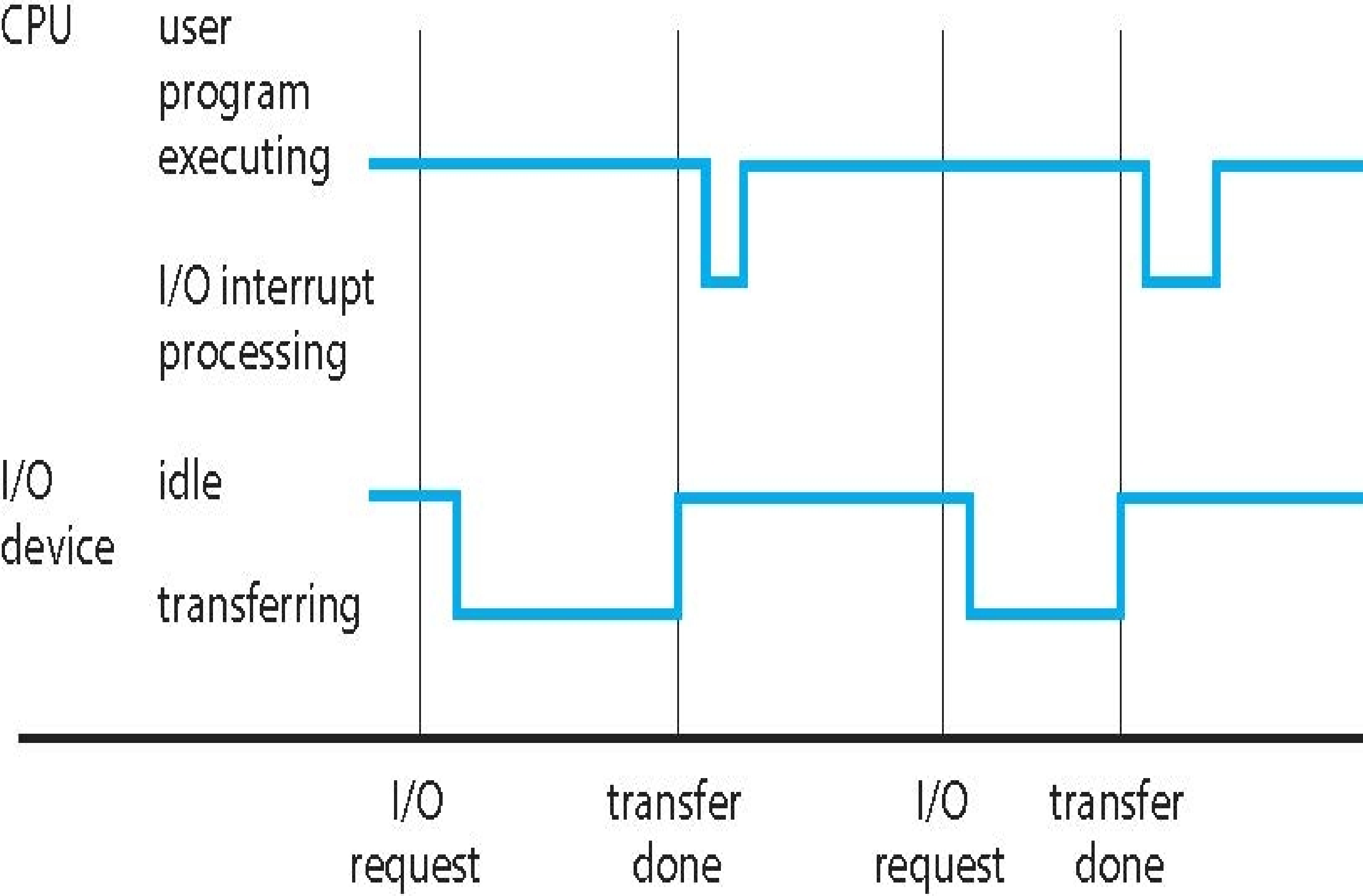
Interrupts

- There are two types of interrupts**:**
  - **Hardware** -- a device may trigger an interrupt by sending a signal to the CPU, usually by way of the system bus.
  - **Software** -- a program may trigger an interrupt by executing a special operation called a **system call.**
- A software-generated interrupt (sometimes called **trap** or **exception**) is caused either by an error (e.g., divide by zero) or a user request (e.g., an I/O request).
- An operating system is **interrupt driven.**

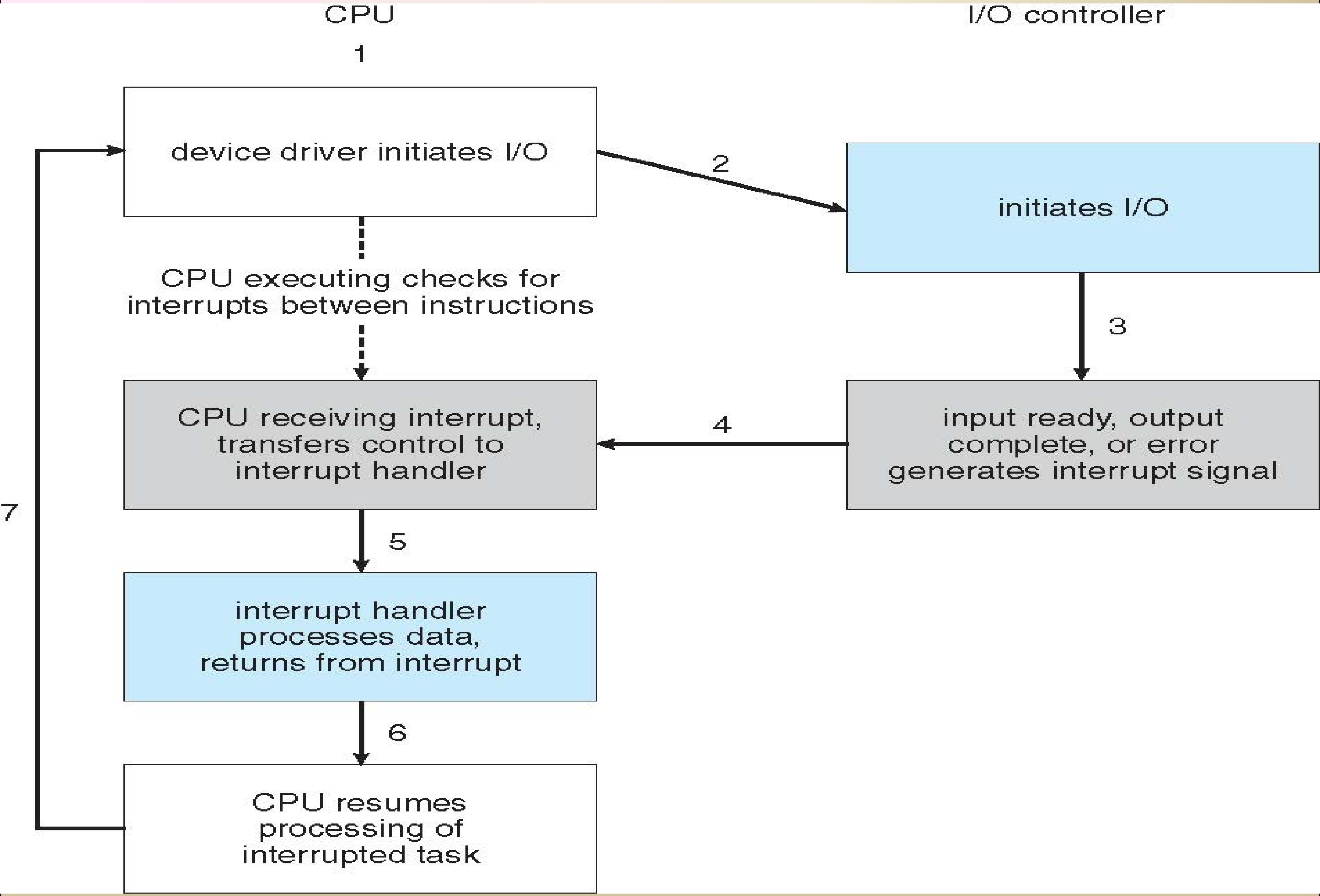
Common Functions of Interrupts

- When an interrupt occurs, the operating system preserves the state of the CPU by storing the registers and the program counter
- Determines which type of interrupt has occurred and transfers control to the interrupt-service routine.
- An interrupt-service routine is a collection of routines (modules), each of which is responsible for handling one particular interrupt (e.g., from a printer, from a disk)
- The transfer is generally through the **interrupt vector**, which contains the addresses of all the service routines
- Interrupt architecture must save the address of the interrupted instruction.

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
4
CPU receiving interrupt, transfers control to interrupt handler
input ready, output complete, or error generates interrupt signal
5
interrupt handler processes data, returns from interrupt
6
CPU resumes processing of interrupted task
7

Storage Definition

- The basic unit of computer storage is the **bit**. A bit can contain one of two values, 0 and 1. All storage in a computer is based on collections of bits.
- A **byte** is 8 bits, and on most computers it is the smallest convenient chunk of storage.
- 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.

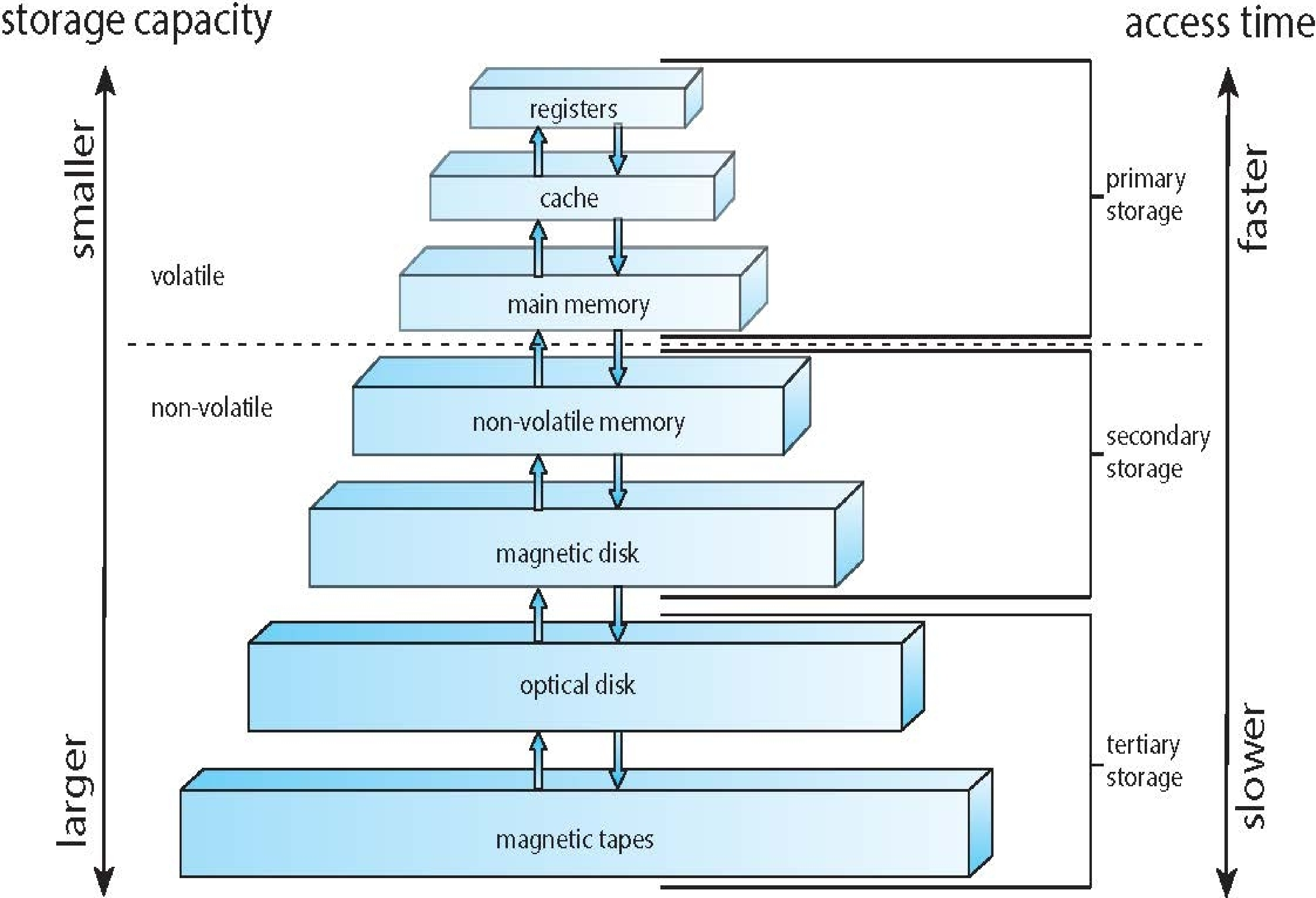
Storage Definition (Cont.)

- 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
  - a **petabyte**, or **PB**, is $1,024^5$ bytes
  - exabyte, zettabyte, yottabyte
- 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 Hierarchy

- Storage systems organized in hierarchy
  - Speed
  - Cost
  - Volatility
- **Caching** – copying information from “slow” storage into faster storage system;
  - Main memory can be viewed as a cache for secondary storage
- **Device Driver** for each device controller to manage I/O
  - Provides uniform interface between controller and kernel

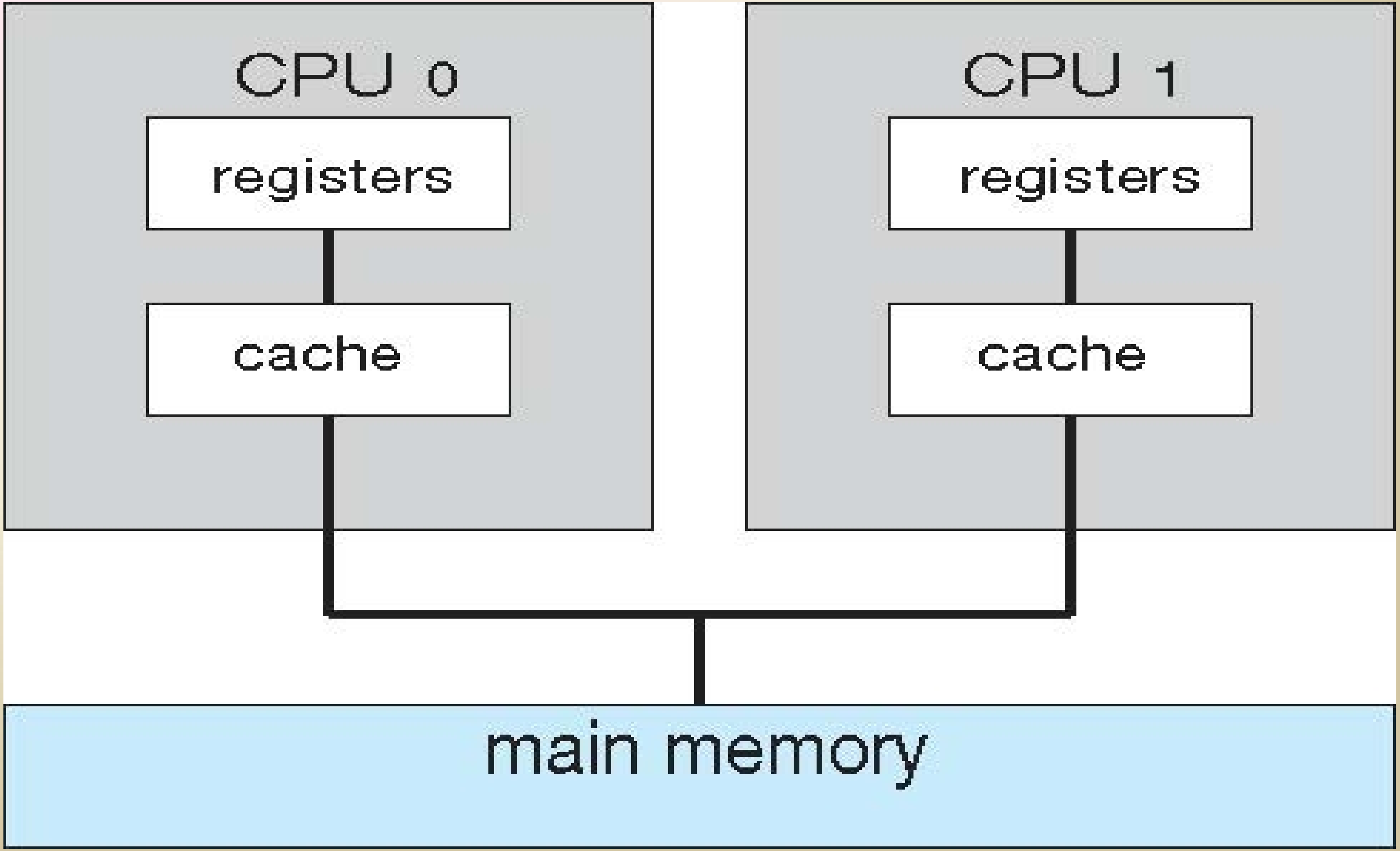
Storage-device hierarchy



Computer-System Architecture

- **Single general-purpose processor**
  - Most systems have special-purpose processors as well
- **Multiprocessors** systems
  - Also known as **parallel systems**, **tightly-coupled systems**
  - Advantages include:
    - **Increased throughput**
    - **Economy of scale**
    - **Increased reliability** – graceful-degradation/fault-tolerance
  - Two types:
    - **Symmetric Multiprocessing** – each processor performs all tasks
    - **Asymmetric Multiprocessing** – each processor is assigned a specific task.

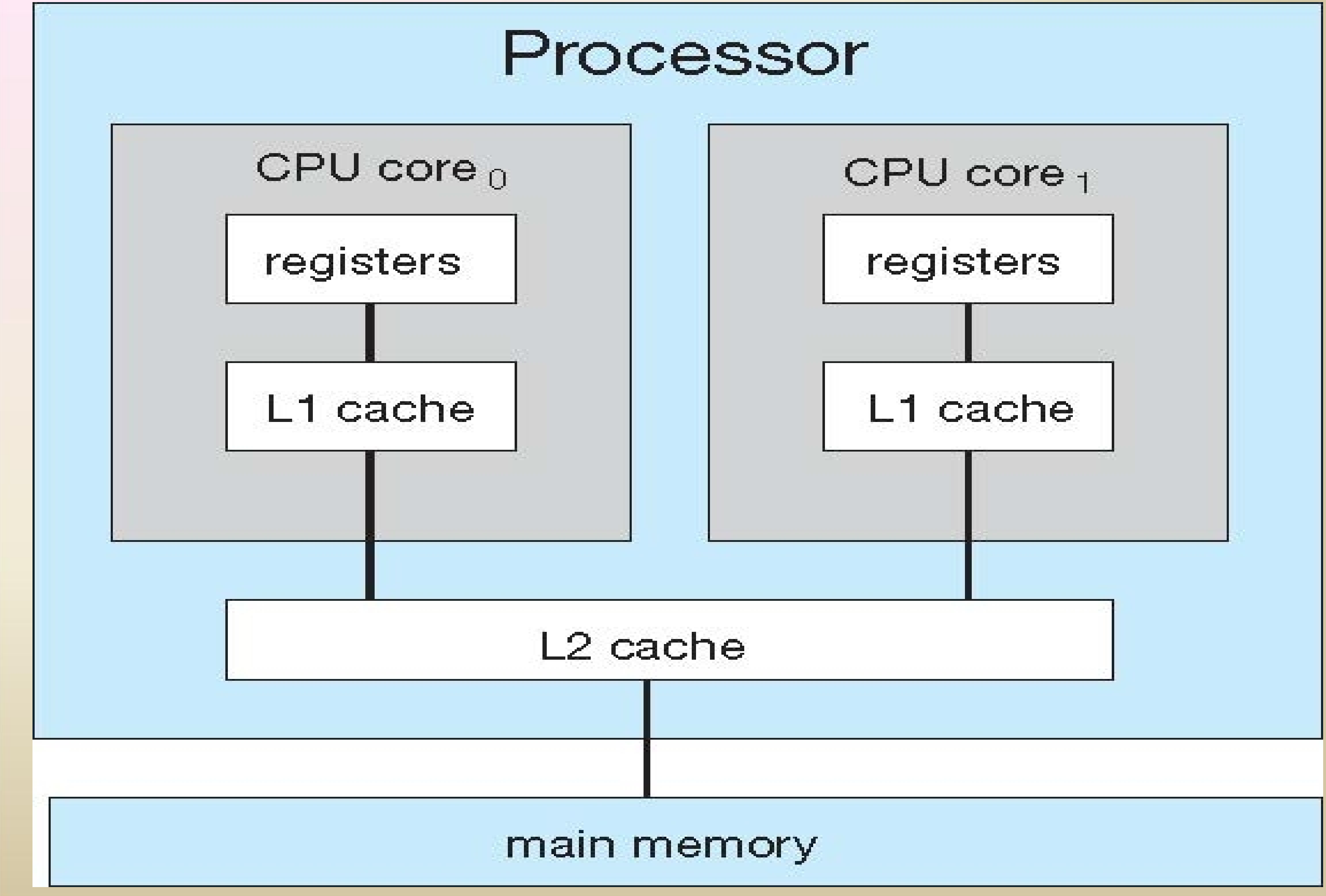
Symmetric Multiprocessing Architecture



Multicore Systems

- Most CPU design now includes multiple computing cores on a single chip. Such multiprocessor systems are termed **multicore**.
- Multicore systems can be more efficient than multiple chips with single cores because:
  - On-chip communication is faster than between-chip communication.
  - One chip with multiple cores uses significantly less power than multiple single-core chips, an important issue for laptops as well as mobile devices.
- Note -- while multicore systems are multiprocessor systems, not all multiprocessor systems are multicore.

A dual-core with two cores placed on the same chip

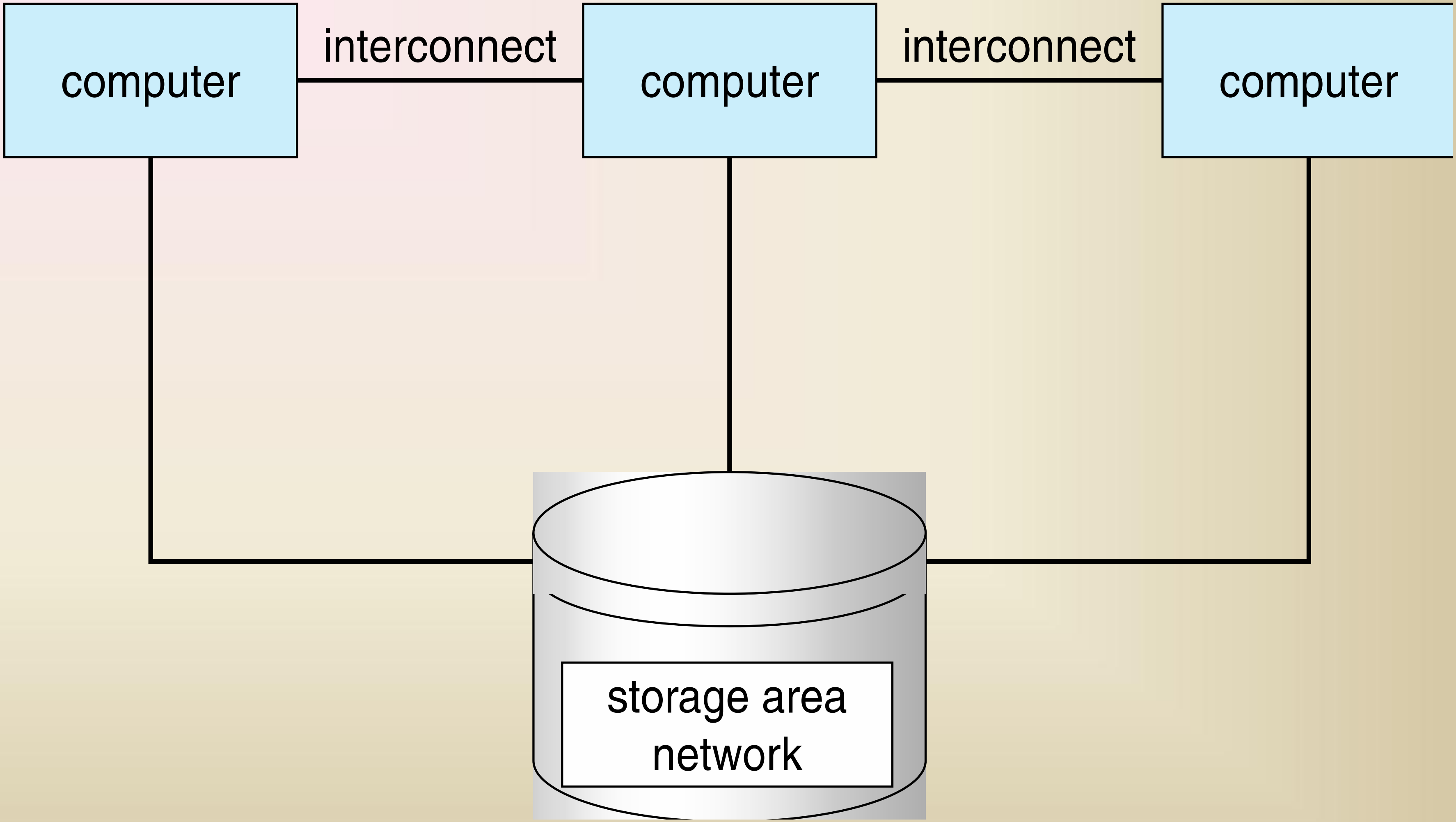


Clustered Systems

Like multiprocessor systems, but multiple systems working together

- Usually sharing storage via a **storage-area network (SAN)**
- Provides a **high-availability** service which survives failures
  - **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



Multiprogrammed System

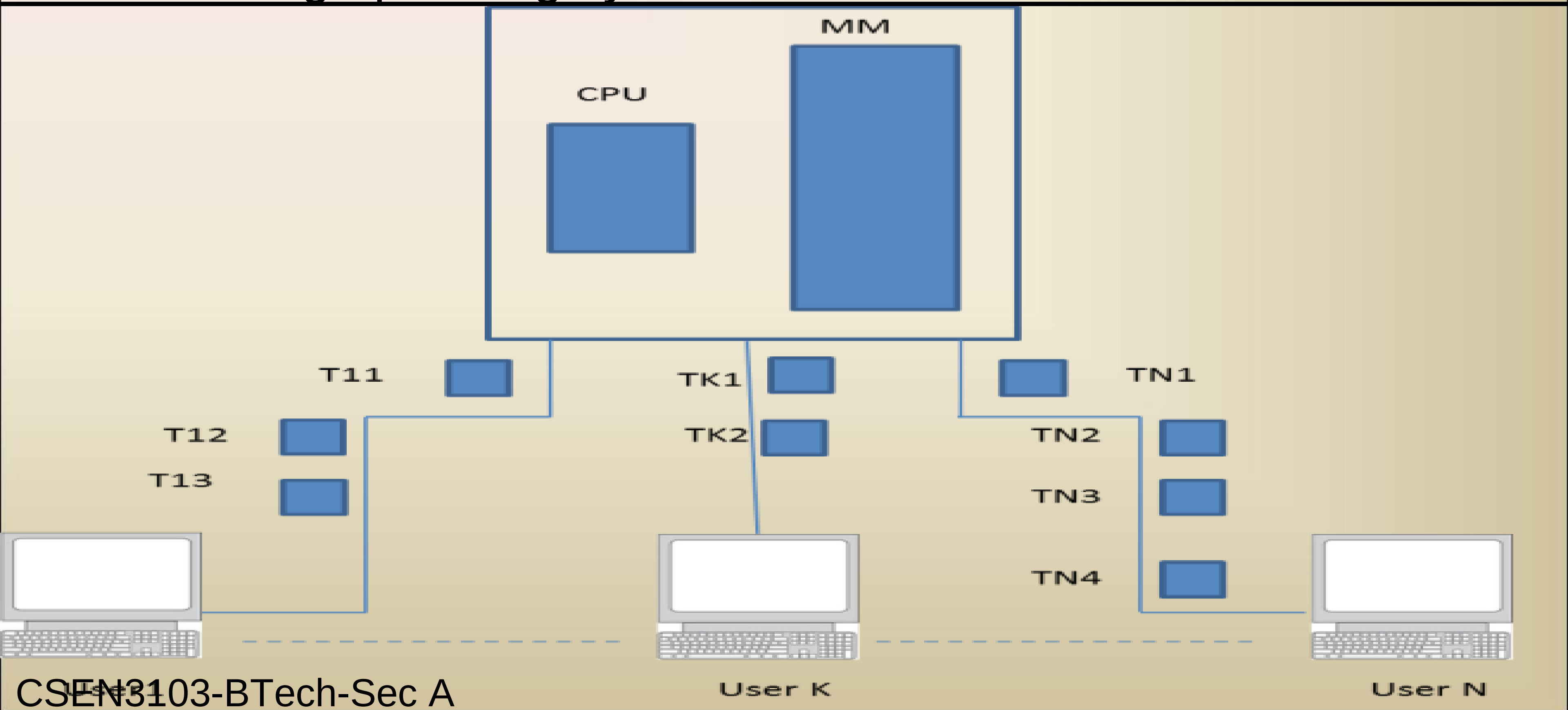
- Single user cannot keep CPU and I/O devices busy at all times
- **Multiprogramming** organizes jobs (code and data) so CPU always has one to execute
- A subset of total jobs in system is kept in memory
- Batch systems:
  - One job selected and run via **job scheduling**
  - When it has to wait (for I/O for example), OS switches to another job
- Interactive systems:
  - Logical extension of batch systems -- CPU switches jobs so frequently that users can interact with each job while it is running, creating **interactive** computing

Objectives: Multiprogramming system is inherently a multi-user system

- Goal of multi programming: Maximum utilisation of CPU time
- Efficient management of main memory
- To achieve the above objectives, we need to answer the following questions:
  - Which program should be brought into main memory?
  - Of the resident programs, which one should be executed next?
- ***The ability of an operating system to accept more than one job from a single user is called multitasking.***

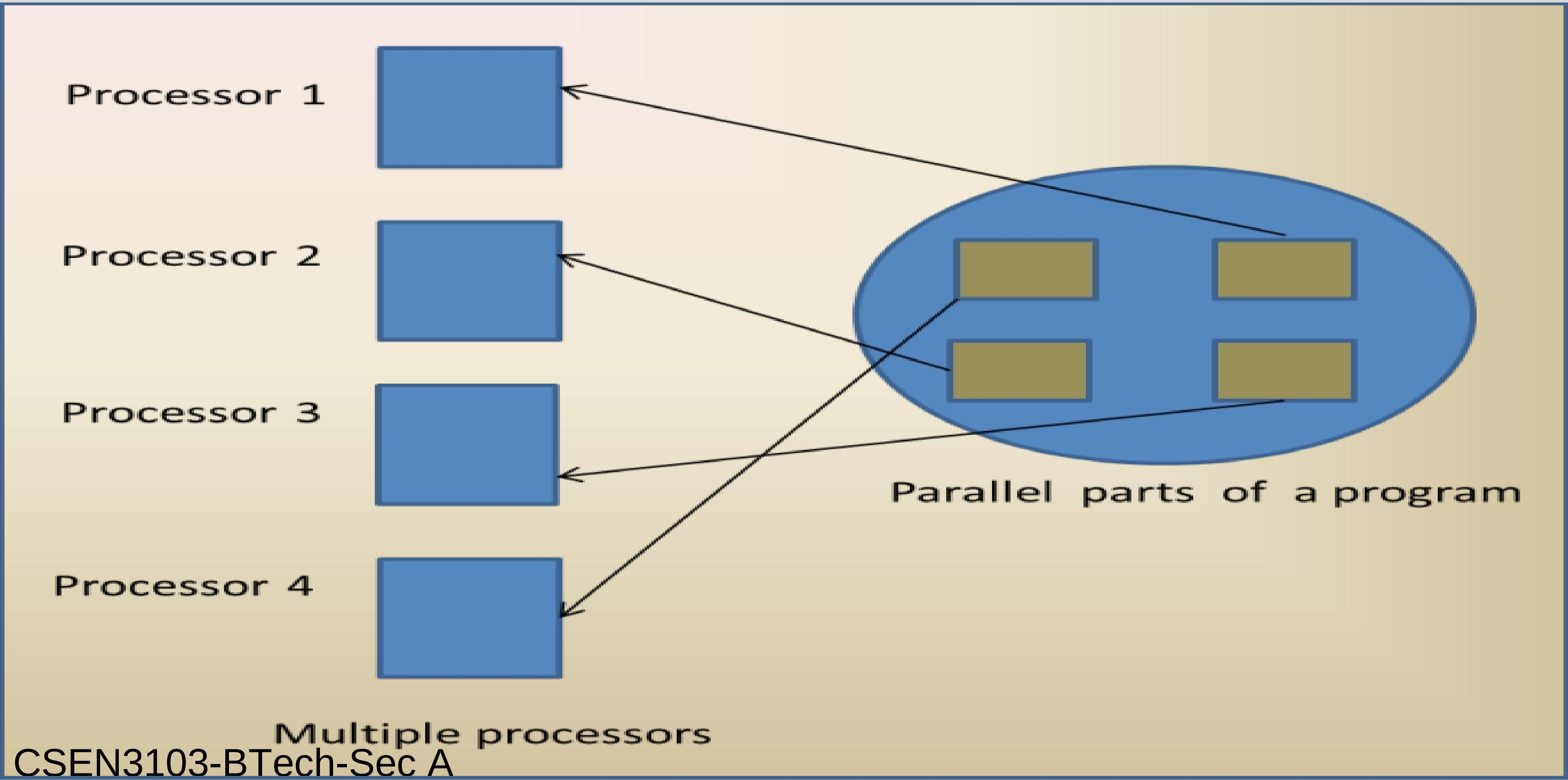
Multitasking System

- **Task**: The user or system job that needs to be performed.
- *The ability of an operating system to accept more than one job from a single user is called **multitasking.***
- Example: UNIX is a multi-user, multiprogramming and multitasking operating system.



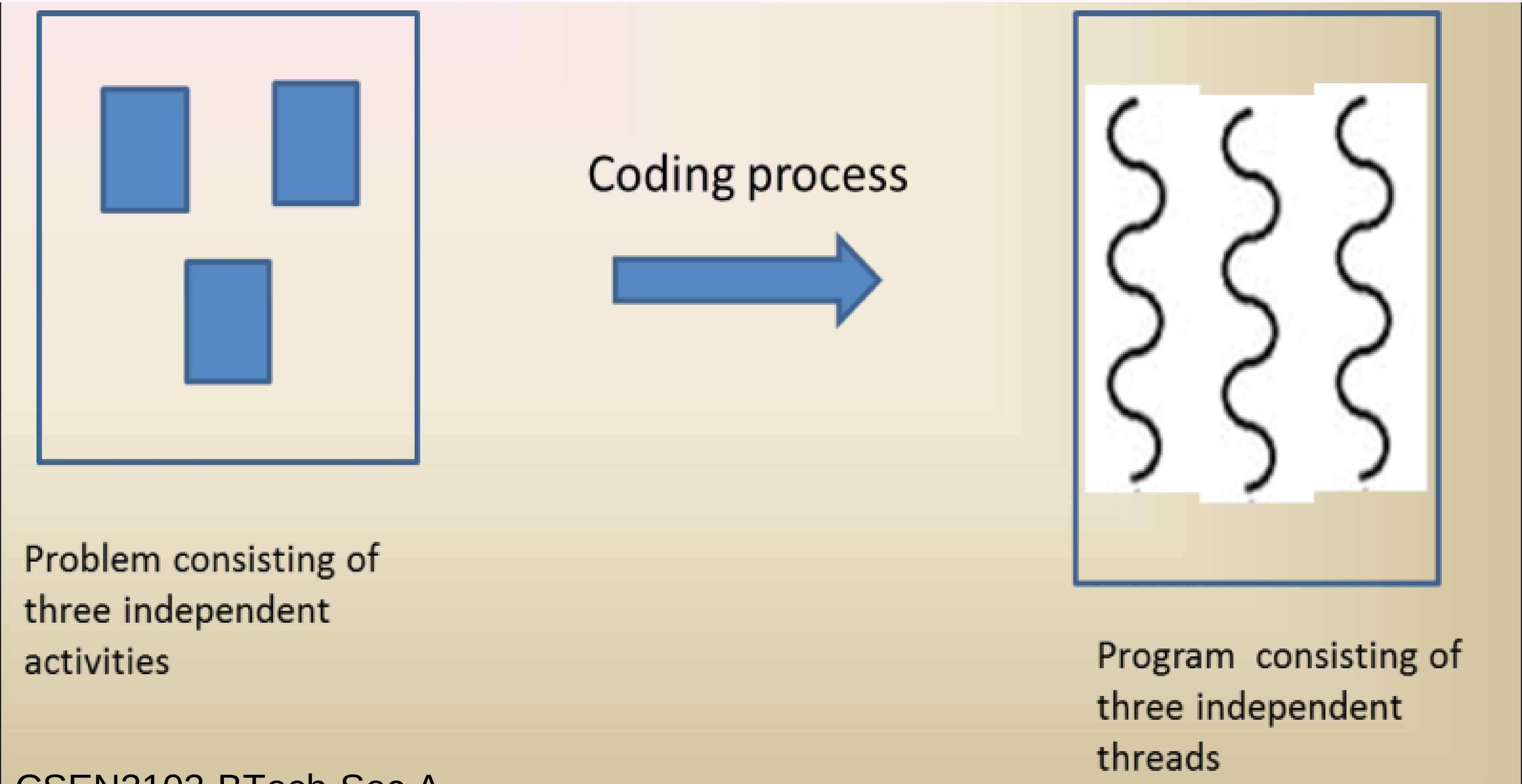
Parallel Processing

There are some algorithms which are inherently parallel in nature. Example, we have quick sort and merge sort dividing the data into partitions that can be processed independently of each other, provided a sufficiently number of processors are available in the computer system.



Multithreading operating system

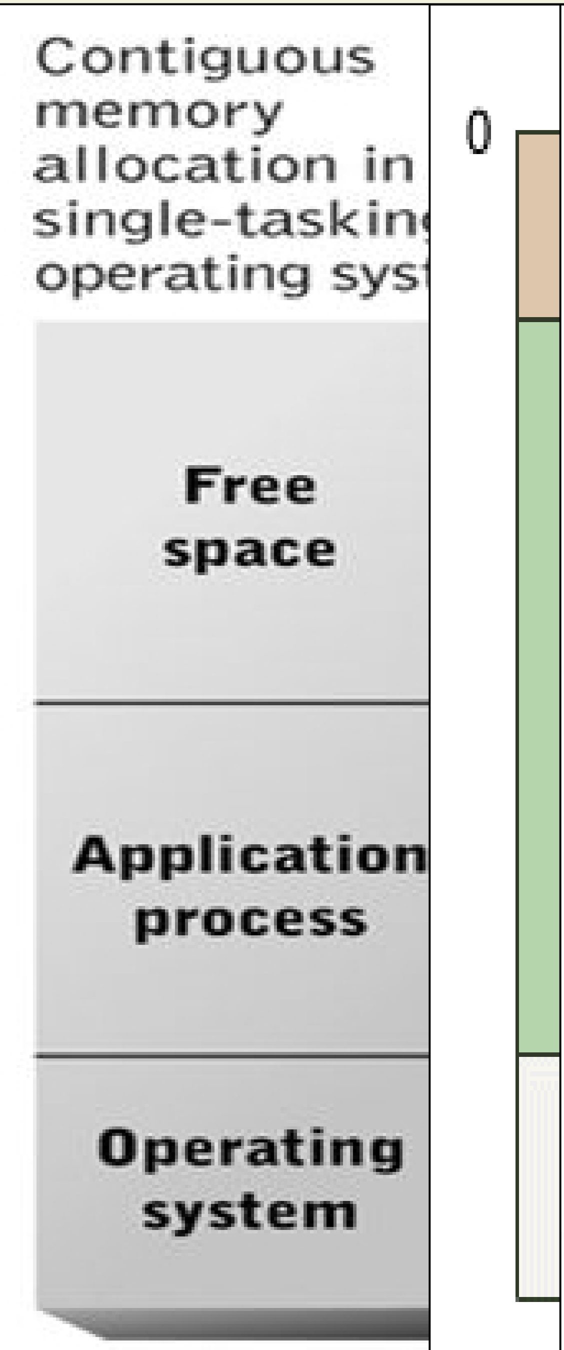
A thread can be precisely defined as a separate path of execution through the code of a program. The operating system that can handle multiple threads of a program is called a multithreading operating system.

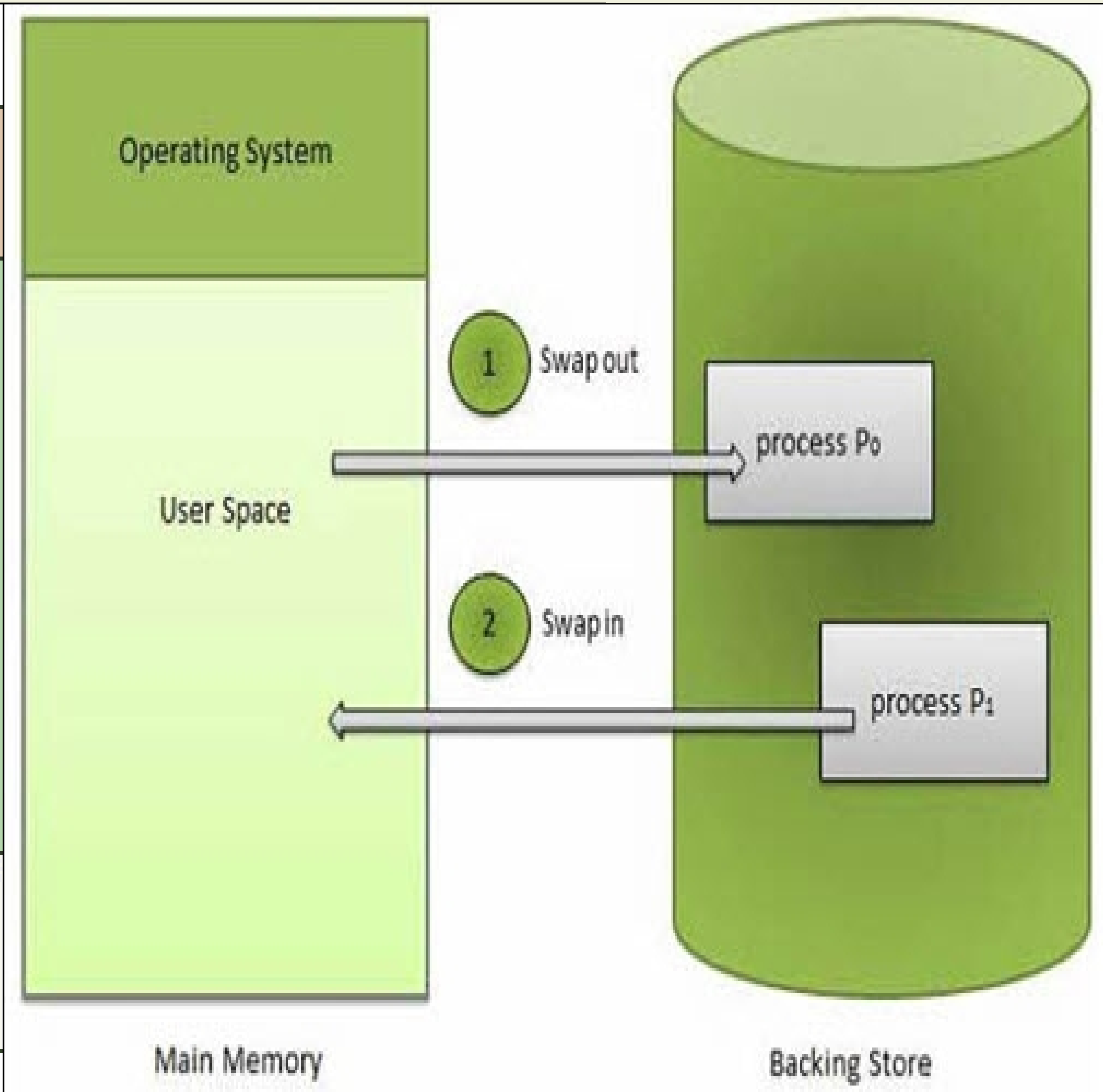


Problem consisting of three independent activities

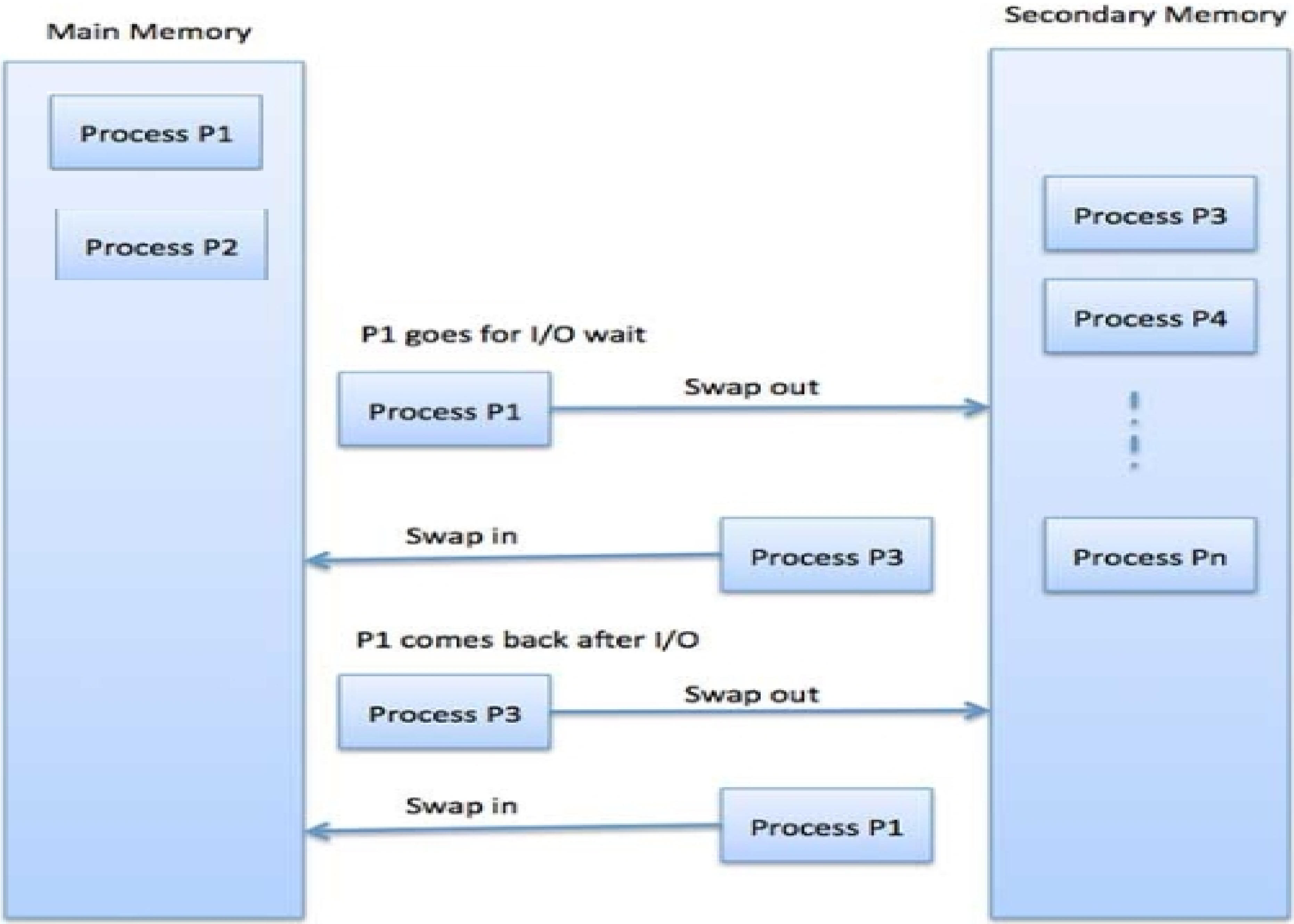
Program consisting of three independent threads

OS - MEMORY MANAGEMENT





OS - MEMORY MANAGEMENT…



START

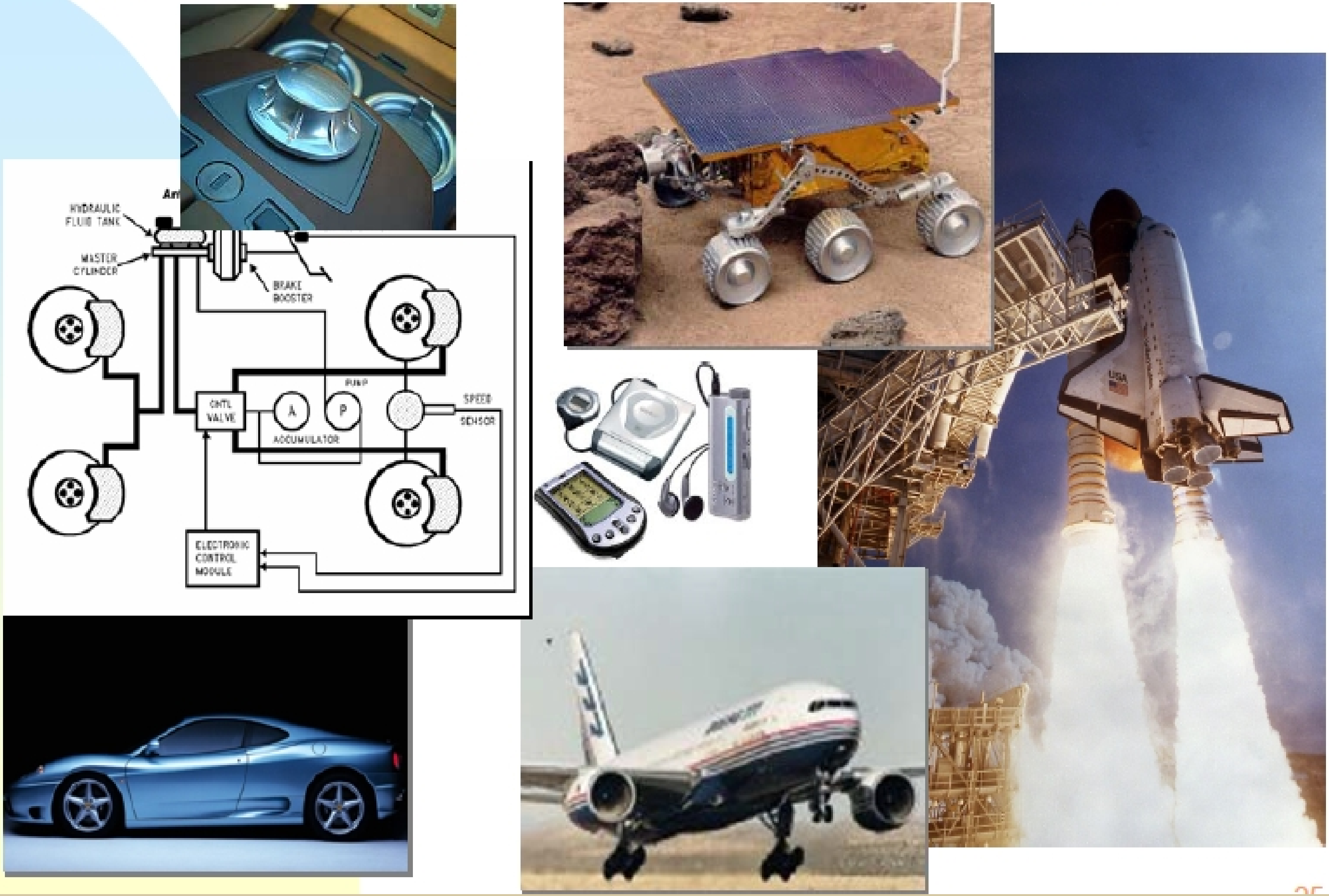
Real-Time Systems

- Definition: A real-time system is a computer system where a timely response by the computer to external stimuli is vital!
- Hard real-time system: The system has failed if a timing constraint, e.g. deadline, is not met.
  - All delays in the system must be bounded.
  - Many advanced features are absent.

Real-Time Systems

- Soft real-time system: Missing a timing constraint is serious, but does not necessarily result in a failure unless it is excessive
  - A critical task has a higher priority.
  - Supported in most commercial OS.
- Real-time means on-time instead of fast

Applications for Real-Time Systems!



Real-Time Systems

- Applications
  - Air traffic control
  - Space shuttle
  - Navigation
  - Multimedia systems
  - Industrial control systems
  - Home appliance controller
  - Nuclear power plant
  - Virtual reality
  - Games
  - User interface
  - Vision and speech recognition (approx. 100 ~ 200ms)
  - PDA, telephone system
  - And more

Open-Source Operating Systems

- Operating systems made available in source-code format rather than just binary **closed-source**
- Counter to the **copy protection** and **Digital Rights Management (DRM)** movement
- Started by **Free Software Foundation (FSF)**, which has “copyleft” **GNU Public License (GPL)**
- Examples include **GNU/Linux** and **BSD UNIX** (including core of **Mac OS X**), and many more
- Can use VMM like VMware Player (Free on Windows), Virtualbox (open source and free on many platforms - http://www.virtualbox.com)
    - Use to run guest operating systems for exploration

Functions of Operating Systems ➔ what’s coming up……

- Process management (processes, threads, cpu scheduling, process synchronization, mutual exclusion, deadlock, semaphores, monitors…..)
- Memory management (MM + VM)
- Processor management (keywords given… concurrency control….)
- Input/Output and file/disk management
- Security

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 features – more OS features (GPS -- Waze)
- 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 – Distributed

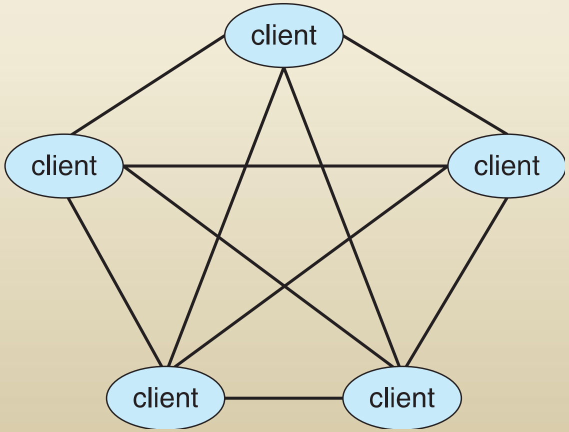
- Collection of separate, possibly heterogeneous, systems networked together
  - **Network** is a communications path, **TCP/IP** most common
    - **Local Area Network** (**LAN**)
    - **Wide Area Network** (**WAN**)
    - **Metropolitan Area Network** (**MAN**)
    - **Personal Area Network** (**PAN**)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

Computing Environments – Client-Server

- 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
  - Each node may 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

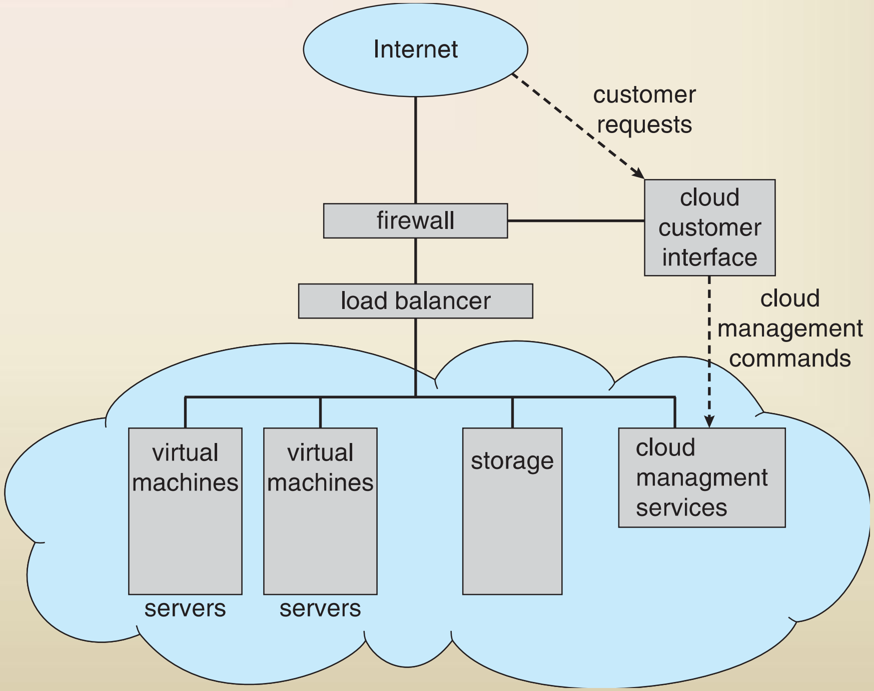


Computing Environments – Cloud Computing

- 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)
  - Infrastructure as a Service (**IaaS**) – servers or storage available over Internet (i.e., storage available for backup use)

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


Internet
customer requests
firewall
cloud customer interface
load balancer
cloud management commands
virtual machines
virtual machines
storage
cloud managment services
servers
servers