Operating Systems

Moumita Patra
July-Nov 2019
Ref: Galvin, Gagne

General Information

- Textbook:

  -> Operating System Concepts 9th edition, Silberschatz, Galvin and Gagne.

- Reference text books:

  -> Operating Systems: Principles & Practice; Anderson and Dahlin.

  -> Modern Operating Systems; Andrew S Tanenbaum.

- Programming assignments will be covered in the associated lab course.
- Course materials will be uploaded on moodle.

Marks Distribution (Tentative)

- Mid-sem -> 30
- Quiz -> 20
- End sem -> 50

Introduction

Operating System

- What is an operating system (OS)?

  -> An OS is a program that manages a computer’s hardware.

  -> A program that acts as an intermediary between a user of a computer and the computer hardware.

OS from different views

- **User-centric**

- A program that acts as an intermediary between a user of a computer and the computer hardware
- Defines an interface for the user to use services provided by the system
- Converts what the hardware gives to what the user wants
- Can hide many details of the hardware that the user does not need to know
- Can even give a very different view of the operating environment to the user than what is actually there

- **System-centric**

➔ Efficiently manages and allocates resources to users

➔ Controls the execution of user programs and operations of I/O devices

➔ Provides isolation/protection between different user programs

- **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

- **Components of a computer system:**

  -> **hardware-** provides basic computing resources

  -> **OS-** controls and coordinates use of hardware among various applications and users

  -> **application programs-** define ways in which the system resources are used to solve computing problems of the users

  -> **users-** people, machines, other computers

Operating System Definition

- No universally accepted/ completely adequate definition.
- Operating systems exist because they offer a reasonable way to solve the problem of creating a usable computing system.
- An approximation is that it includes everything that a vendor ships when an OS is ordered. The features may vary widely across systems.
- More commonly- OS is the one program running at all times on the computer called **kernel**.
- Two types of other programs are:

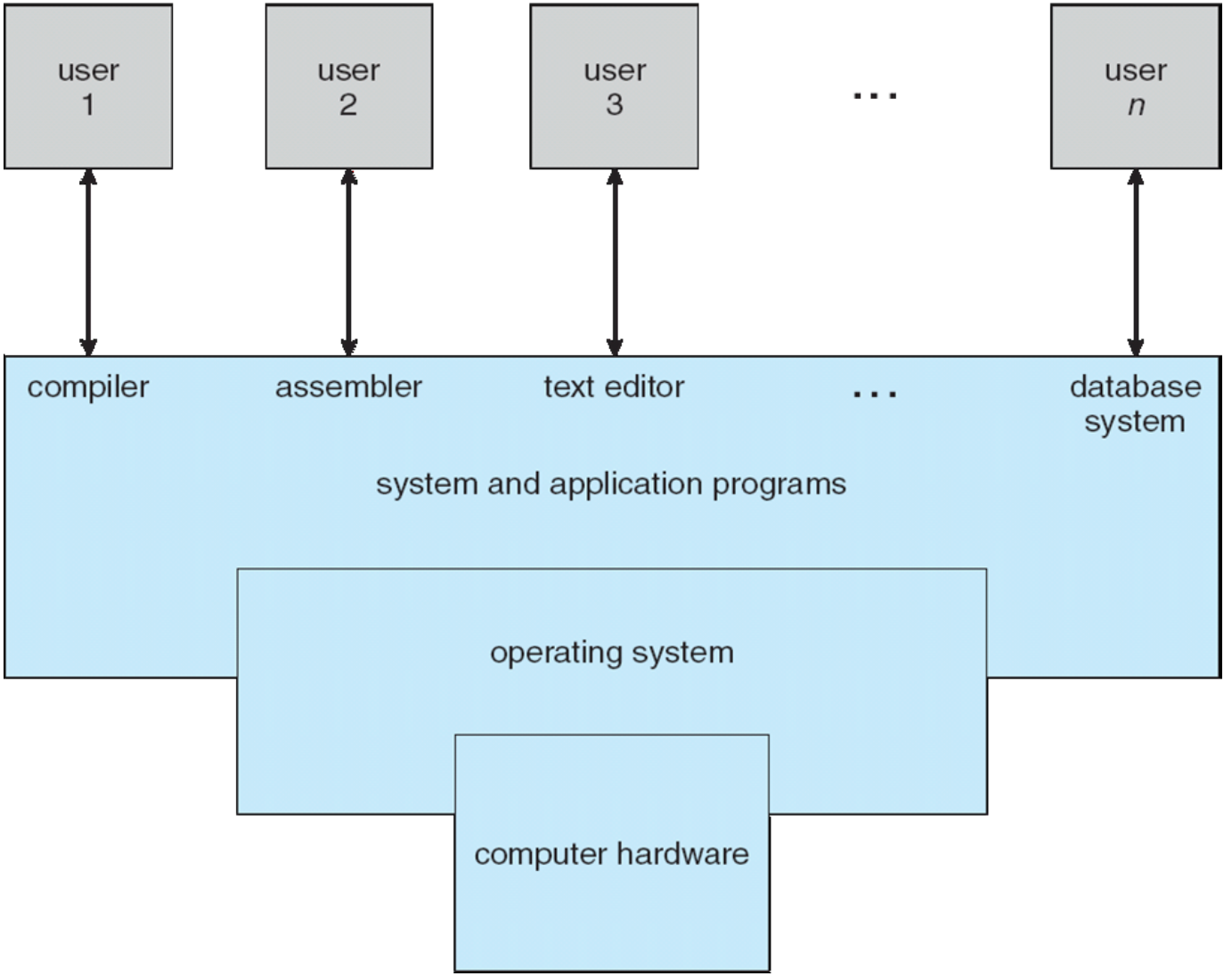
➔ **System programs**- associated with the OS but not necessarily part of the kernel

➔ **Application programs**- include all programs not associated with the operation of the system

NOTE:

- Mobile operating systems often include not only a core kernel but also **middleware**
- **Middleware-** a set of software frameworks that provide additional services to application developers.
- Example- Both iOS and Android feature a core kernel along with middleware that supports databases, multimedia, and graphics (to name a few).

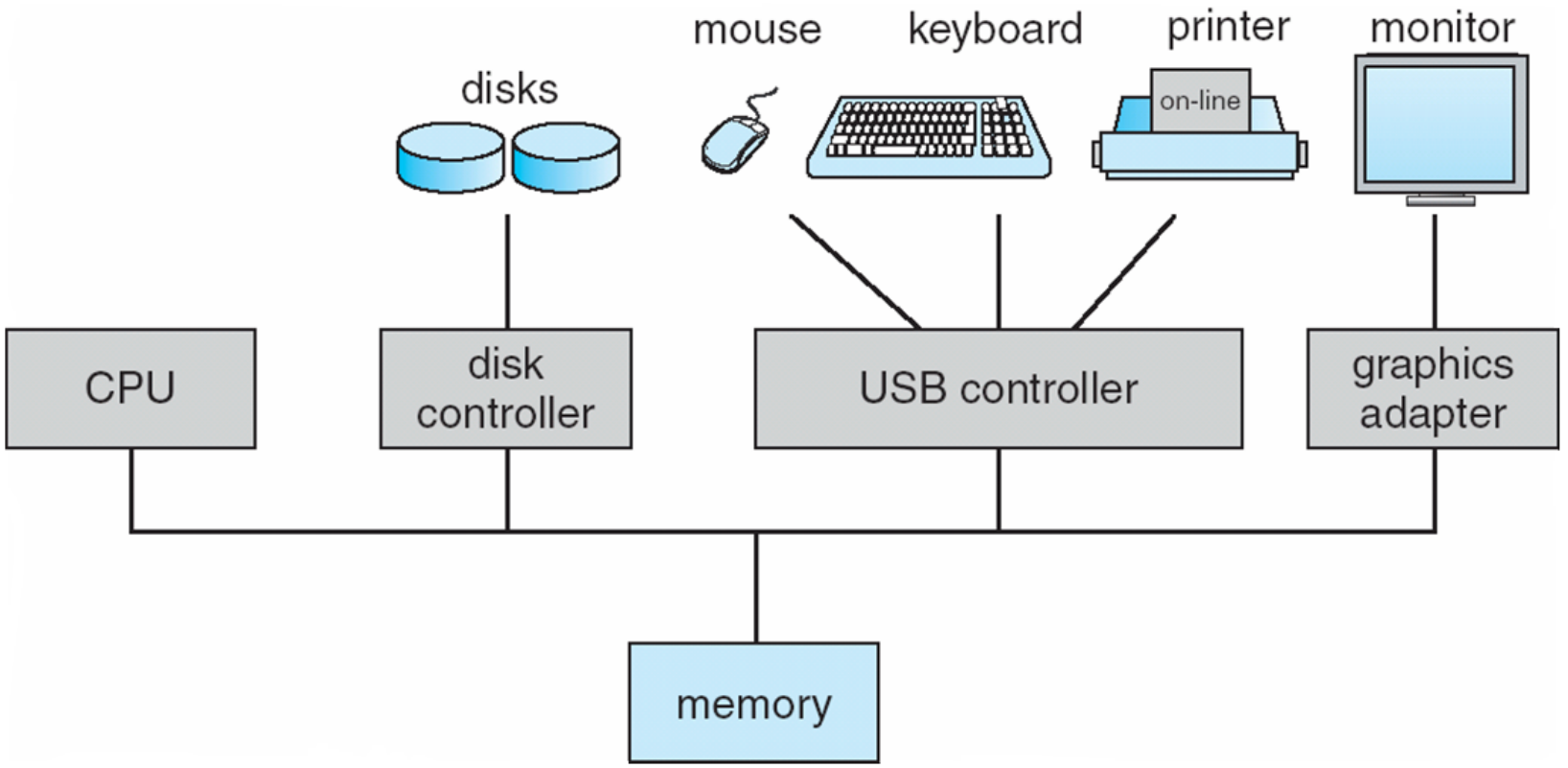
Components of a computer system



Computer Startup

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

Computer System Organization



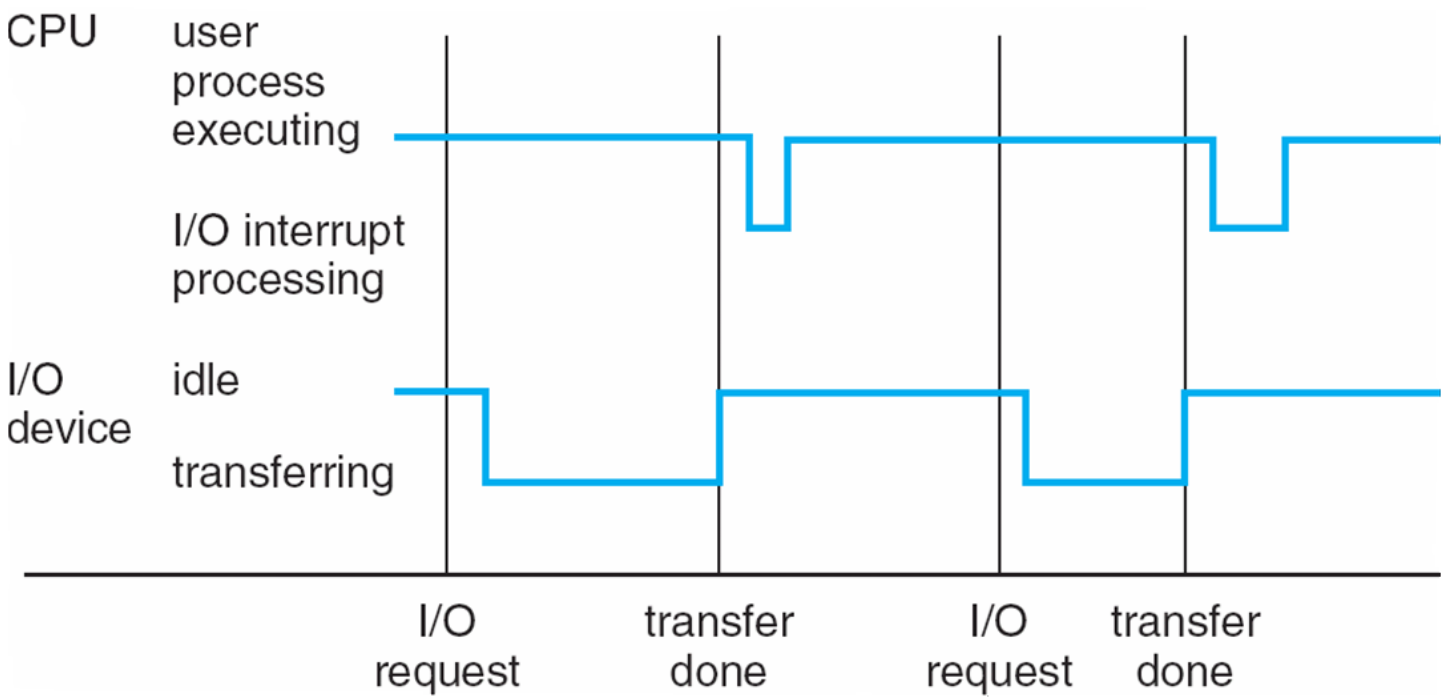
Computer System Operation

- I/O devices and the CPU can execute concurrently
- Each device controller is in charge of a particular device type
- Each device controller has a local buffer
- CPU moves data from/to main memory to/from local buffers
- I/O is from the device to local buffer of controller
- Device controller informs CPU that it has finished its operation by causing an interrupt

Common Functions of Interrupts

- A signal, generated to get the attention of the CPU. Usually generated when I/O is required.
- 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**

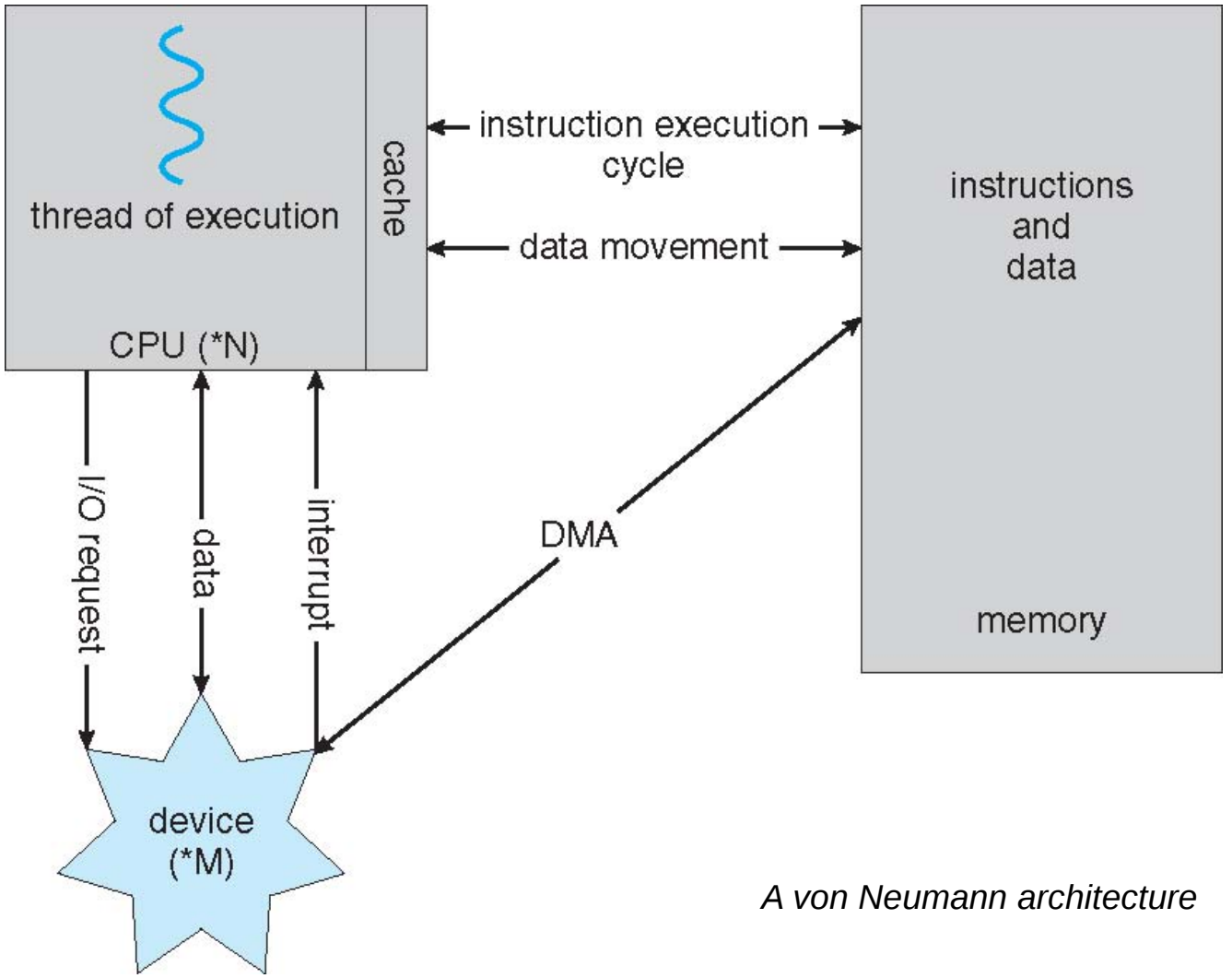
Interrupt Timeline



I/O Structure

- After I/O starts, control returns to user program only upon I/O completion
  - Wait instruction idles the CPU until the next interrupt
  - Wait loop (contention for memory access)
  - At most one I/O request is outstanding at a time, no simultaneous I/O processing
- After I/O starts, control returns to user program without waiting for I/O completion

How a modern computer works



A von Neumann architecture

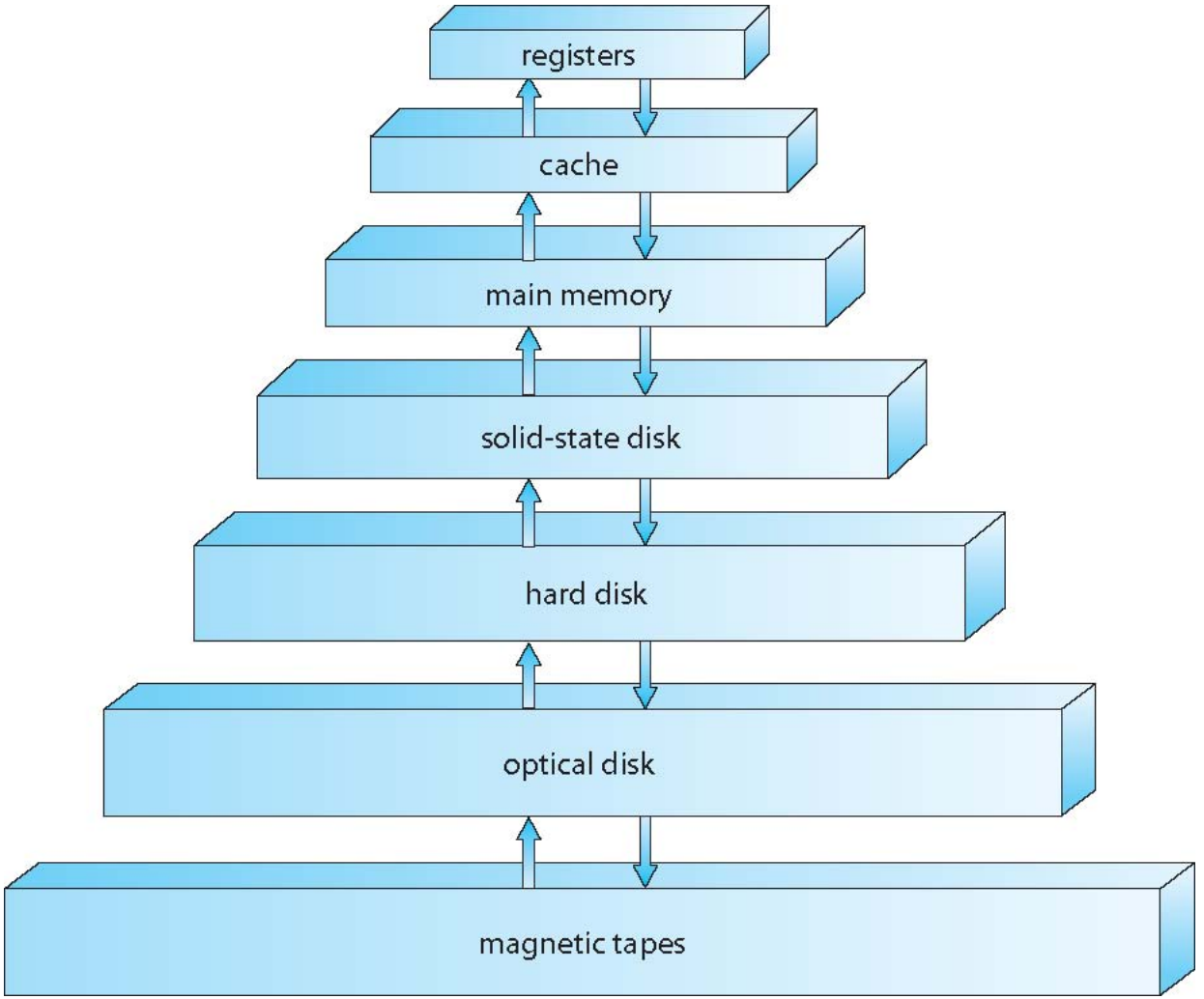
Storage Structure

- 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

Storage Hierarchy

- Storage systems organized in hierarchy
  - Speed
  - Cost
  - Volatility
- **Caching** – copying information 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

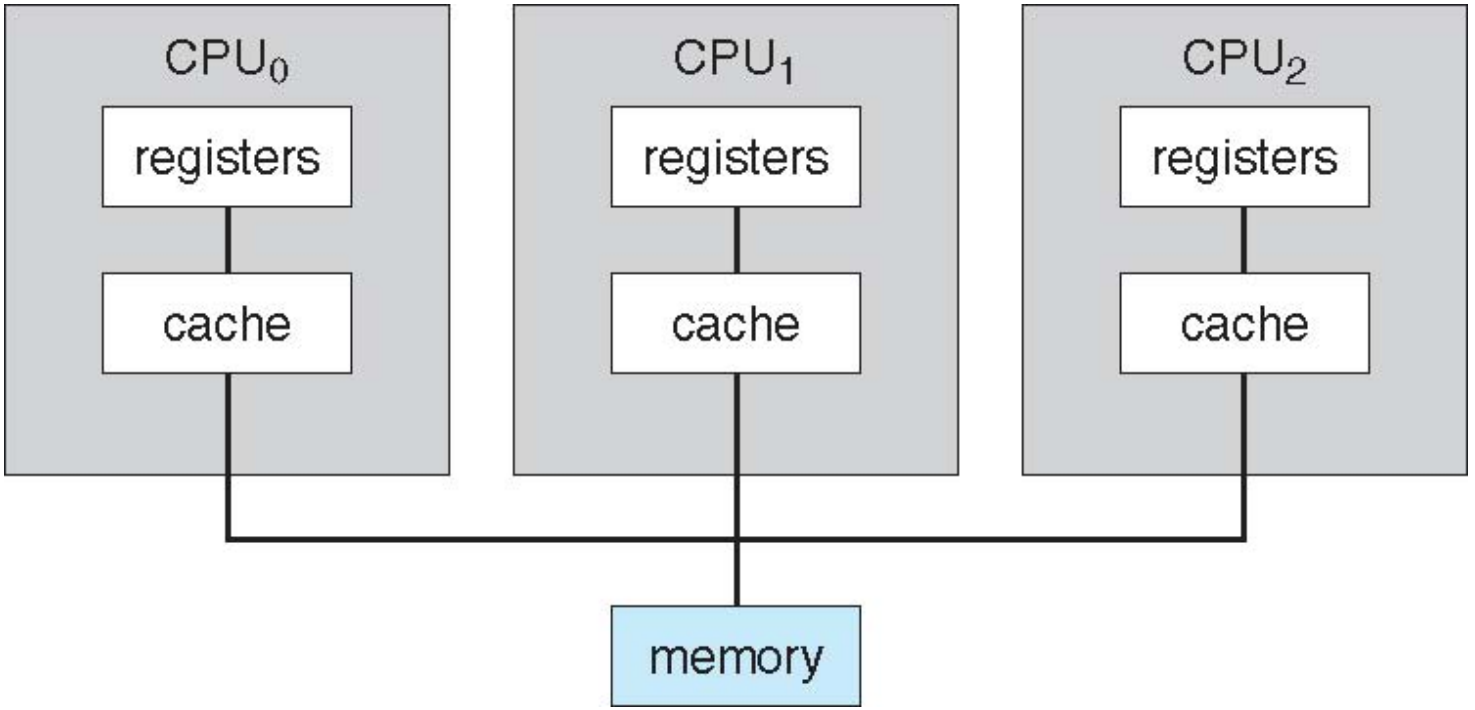
- Single and multi-processor systems
- Multi-processor systems have gained importance over time
- Advantages :

  1. Increased throughput
  2. Economy of scale
  3. Increased reliability

Computer-System Architecture

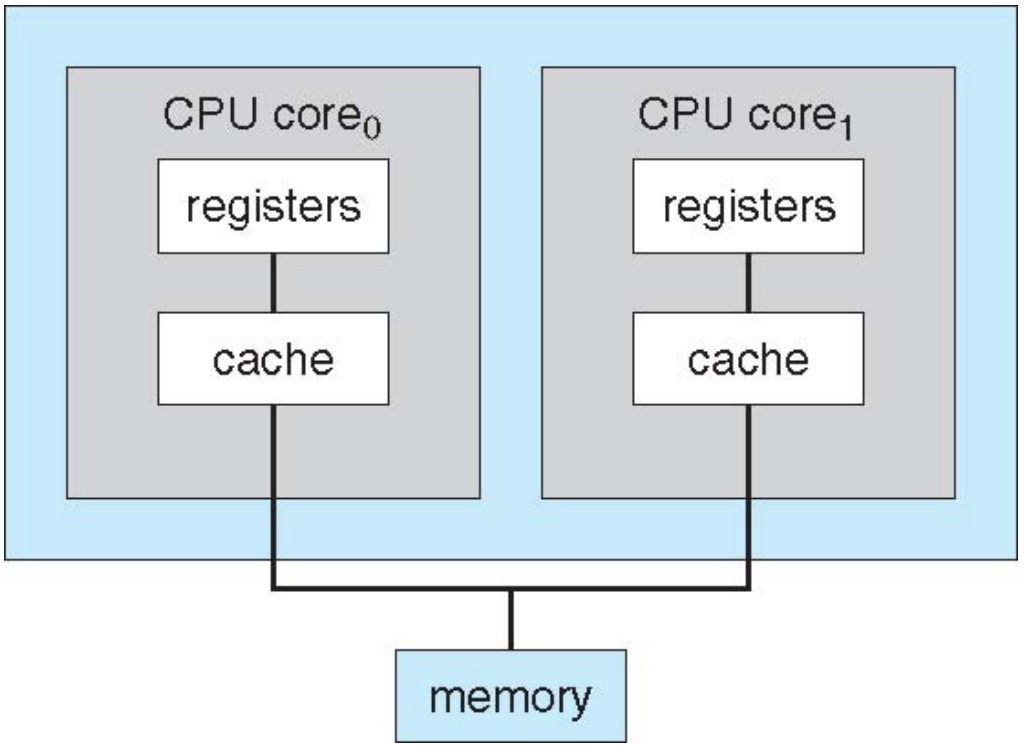
- Most systems use a single general-purpose processor
- **Multiprocessor** systems growing in use and importance
  - Also known as **parallel systems**, **tightly-coupled systems**
  - Advantages include:
    1. **Increased throughput**
    2. **Economy of scale**
    3. **Increased reliability** – graceful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

Symmetric Multiprocessing Architecture



A Dual-Core Design

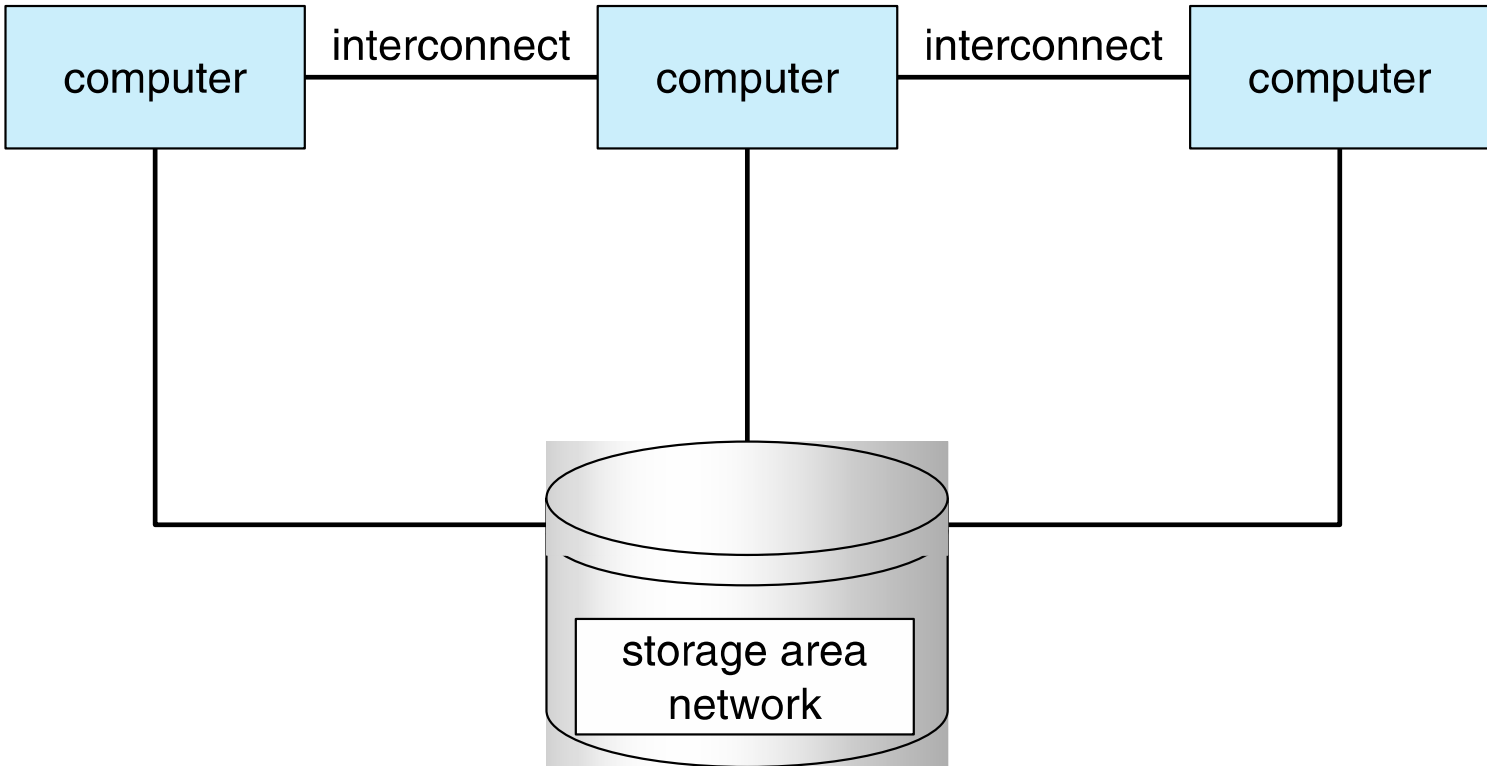
- Multicore- Multiple computing cores on a single chip



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



Operating System Structure

- One of the most important aspects of operating systems is to multiprogram.
- Multiprogramming needed for efficiency
  - Efficient CPU and I/O utilization
  - Jobs are selected via job scheduling
  - When a program waits, OS switches to another job
- Time sharing (multitasking)- CPU switches jobs frequently so that users can interact with each job while it is running, creating interactive computing.
  - Each user has at least one program executing in memory process
  - CPU scheduling needed if several jobs ready to run at the same time.

Types of systems

- **Batch systems**

  Multiple jobs, but only one job in memory at one time and executed (till completion) before the next one starts

- **Multiprogrammed batch systems**

  Multiple jobs in memory, CPU is multiplexed between them

- **Time-sharing systems**

  Multiple jobs in memory and on disk, CPU is multiplexed among jobs in memory, jobs swapped between disk and memory

  Allows interaction with users

- **Personal Computers**

  -> Dedicated to single user at one time
- **Multiprocessing Systems**

  -> More than one CPU in a single machine to allocate jobs to

  -> Multicore architectures
- Other paraller systems, distributed systems, clusters...

  -> Different types of systems with multiple CPUs/ Machines
- **Real time systems**

  -> Systems to run jobs with time guarantees
- Other types

  -> Embedded systems, mobiles/smartphones, ...

- OS design depends on the type of system it is designed for
- Challenges today-

➔ Very wide variety of systems

➔ From small embedded systems with low memory and storage to very large systems with hundreds of thousands of machines with large distributed storage

➔ Increasing number of cores per processor and processors per machine

Resources managed by OS

- **Physical**

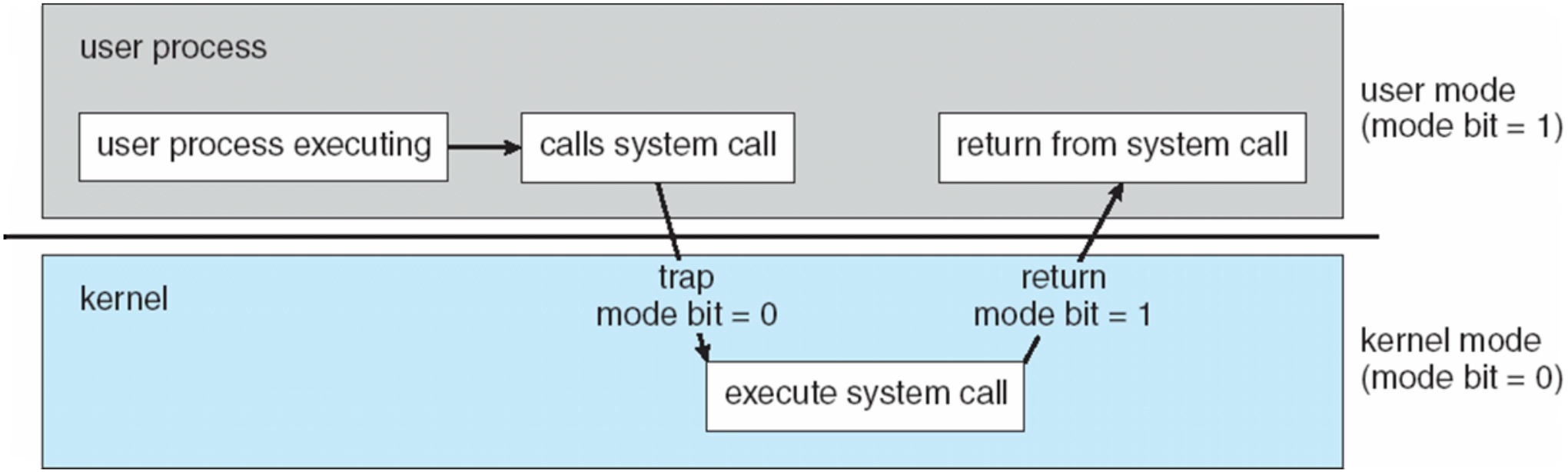
  CPU, memory, disk, I/O devices like keyboard, monitor, printer

- **Logical**

  Process, file ...

Operating System Operations

- In order to ensure proper execution of the OS, it is necessary to distinguish between **system code** and **user-defined code**.
- 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 mode (1) or kernel mode (0)**
  - When the computer system is executing on behalf of a user application, the system is in **user mode**
  - When a user application requests a service from the OS (via a system call), the system must transition from user to **kernel mode**


user process
user process executing
calls system call
return from system call
user mode
(mode bit = 1)
kernel
trap
mode bit = 0
return
mode bit = 1
execute system call
kernel mode
(mode bit = 0)

User-centric view

- **System calls**- programmatic way in which a computer program requests a service from the kernel of the OS it is executed on.
- **Command interpreter** (not strictly a part of an OS)

Main components of an OS

- **Resource-centric view**
  - Process management
  - Main memory management
  - File management
  - I/O system management
  - Secondary storage management
  - Security and protection system
  - Networking

Process Management

- A **process** is a program in execution
- A process is the unit of work in a system
- A system consists of a collection of processes, some of which are OS processes and rest are user processes
- Needs certain resources to accomplish its task

  -> CPU time, memory, files, I/O devices...
- **OS responsibilities**

  -> Process creation and deletion

  -> Process suspension and resumption

  -> Mechanisms for process synchronization and interprocess communication

Main memory management

- Main memory is a large array of bytes. Each byte has its own address.
- It is generally the only large storage that the CPU is able to address and access directly
- For a program to be executed, it must be mapped to absolute adresses and loaded into memory
- **OS responsibilities-** 

➔ Keep track of which parts of memory are currently being used and by whom

➔ Decide which processes to load when memory space becomes available

➔ Allocate and deallocate memory space as needed

➔ Protect memory of one process from another

File management

- To make the computer system convenient for users, the OS provides a uniform, logical view of information storage.
- The OS abstracts from the physical properties of its storage devices to define a logical storage unit called the **file**.
- **OS responsibilities-**

➔ File creation, deletion, modification

➔ Directory creation, deletion, modification

➔ Support of primitives for manipulating files and directories

➔ Mapping files onto secondary storage

➔ File backup on stable (non volatile) storage media

I/O system management

The I/O system consists of:

- A buffer-caching system
- Device driver interface
- Drivers for specific hardware devices

Secondary storage management

- Most modern computer systems use disks as the principle on-line storage medium, for both programs and date
- **OS responsibilities**
  - Free space management
  - Storage allocation
  - Disk scheduling

Security and protection system

- Protection refers to a mechanism for controlling access by programs, processes, or users to both system and user resources
- The protection mechanism must:

➔ Distinguish between authorized and unauthorized usage

➔ Specify the controls to be imposed

➔ Provide a means of enforcement