

Operating Systems

Prepared by Dr. Alaa Eldeen Ahmed

IT263 / CYB510: Ch1: Introduction



- Objectives:
 - To provide coverage of basic computer system organization
 - To provide a grand tour of the major operating systems components

- Topics:
 - What is an Operating Systems
 - Computer-System Organization
 - Computer-System Architecture
 - Operating-System Structure
 - Operating-System Operations

What is an Operating System?



- A program that acts as an intermediary between a user of a computer 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 is an Operating System?



- A program that acts as an intermediary between a user of a computer 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 is an Operating System?



- A program that acts as an intermediary between a user of a computer 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



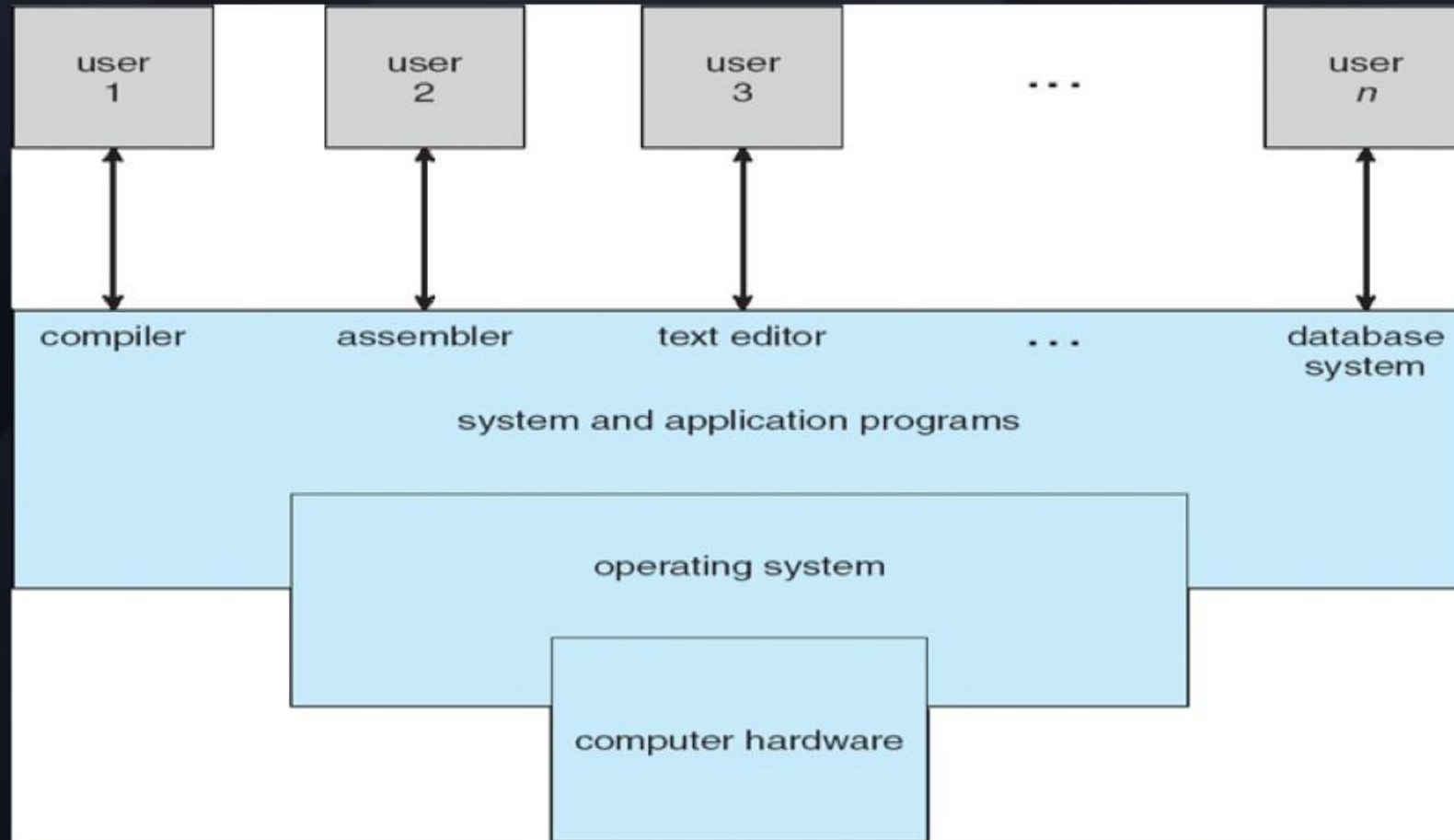
Computer System Structure

- 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

Computer System Structure

- **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

Four Components of a Computer System



Operating System Definition

- **OS is a resource allocator:** The main functions of a resource allocator consist of The following:
 1. **Allocation of Resources:** The resource allocator is in charge of allocating resources to tasks in a way that optimizes system speed and efficiency, including CPU time, memory, I/O devices, and network bandwidth.
 2. **Management of Resources:** It keeps track of the availability of resources in the system and uses preset mechanisms or algorithms to decide how to allocate them among competing programs tasks.

Operating System Definition

- 3. Scheduling of Tasks:** is in charge of scheduling activities to run on the available resources as efficiently as possible, taking into consideration several factors such as priority, fairness and deadlines.
- 4. Preventing overuse of Resource Limits:** It can stop programs from using up too many resources by enforcing the resource limits or restriction defined by the operating system

Operating System Definition

- OS is a **control program**: The main functions of a resource control Program includes the following:
 1. **Controls execution** of programs to prevent errors and improper use of the computer
 2. **Controls access to** system resources and protects the system from unauthorized access of malicious users.

Operating System Definition

3. **Controls the interaction** between software and hardware devices (using Device Drivers)
4. **Controls Files manipulation** such as creation, deletion, and of files and directories on storage devices.
5. **Controls the allocation and deallocation** of memory space for programs.

Operating System Definition (cont)

- **A more common definition** is that the operating system is the one program running at all times on the computer and usually called **The kernel**.
- Along with the kernel, there are two other types of programs:
 1. System programs, which are associated with the operating system but are not part of the kernel such as File browsers, login program., and system services manger.
 2. Application programs, which include all programs not associated with the operation of the system such as web browsers, media players and calculators.

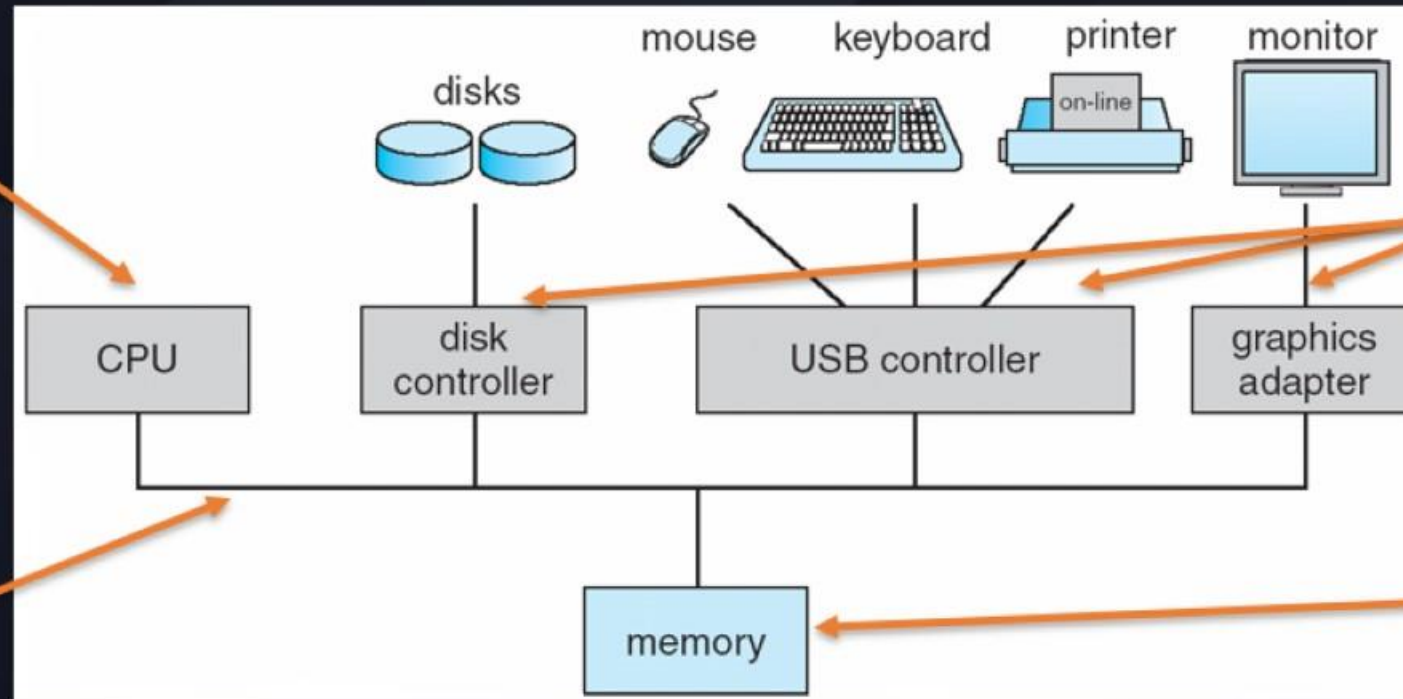
- Some basic knowledge of the structure of Computer System is mandatory to understand the work of operating systems.
Operation
- A modern general purpose Computer system consists of One or more CPUs, device controllers connect through common bus providing access to shared memory
- Concurrent execution of CPUs and devices competing for memory cycles (all programs should loaded into memory to execute)

Chapter 01

Computer System Organization

CPU is the processing unit that is responsible of doing computations and calculations. System may have more than one CPU.

All devices and shared memory are connected Via a common Bus



Controllers are responsible for defining the way of how these devices work.

All programs should be loaded into main memory to execute. Main memory is finite and has a limited size

Computer-System Operation

- All devices are required to share memory to execute correctly, So a memory controller exists to ensure each of these devices get the proper and right access of memory
- 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*

Important Terms

- **Bootstrap Program**

1. The first piece of code that runs when the Computer system starts up, It is stored in ROM.
2. It performs a series of tasks to initialize the computer system and prepare it for operation such as: loading the rest of operating system by locating and Loading kernel into main memory.

Once the operating system is loaded and became running, it takes over and continues the boot process, loading drivers and other system components as needed.

Some Important Terms

- **Interrupt**

- The occurrence of event is usually signaled by Interrupt from hardware or software.
- Hardware may trigger an interrupt at any time by sending a signal to the CPU, usually by the way of system bus.

- **System Call:**
 - Software may trigger an interrupt by executing a special operation called System Call.
 - System call is the programmatic way in which a computer program request a service from the kernel of the operating system

Some Important Terms

- When the CPU is interrupted, it stops what it is doing and immediately transfers execution to fixed location.
 - The fixed location usually contains the starting address where the service Routines of the interrupt is located.
- The Interrupt service Routines executes.
- On completion, the CPU resumes the interrupted computation.

- **Main memory:** is the only large storage media that the CPU can access directly
- **Secondary storage:** is an extension of main memory that provides large nonvolatile storage capacity

Storage Hierarchy

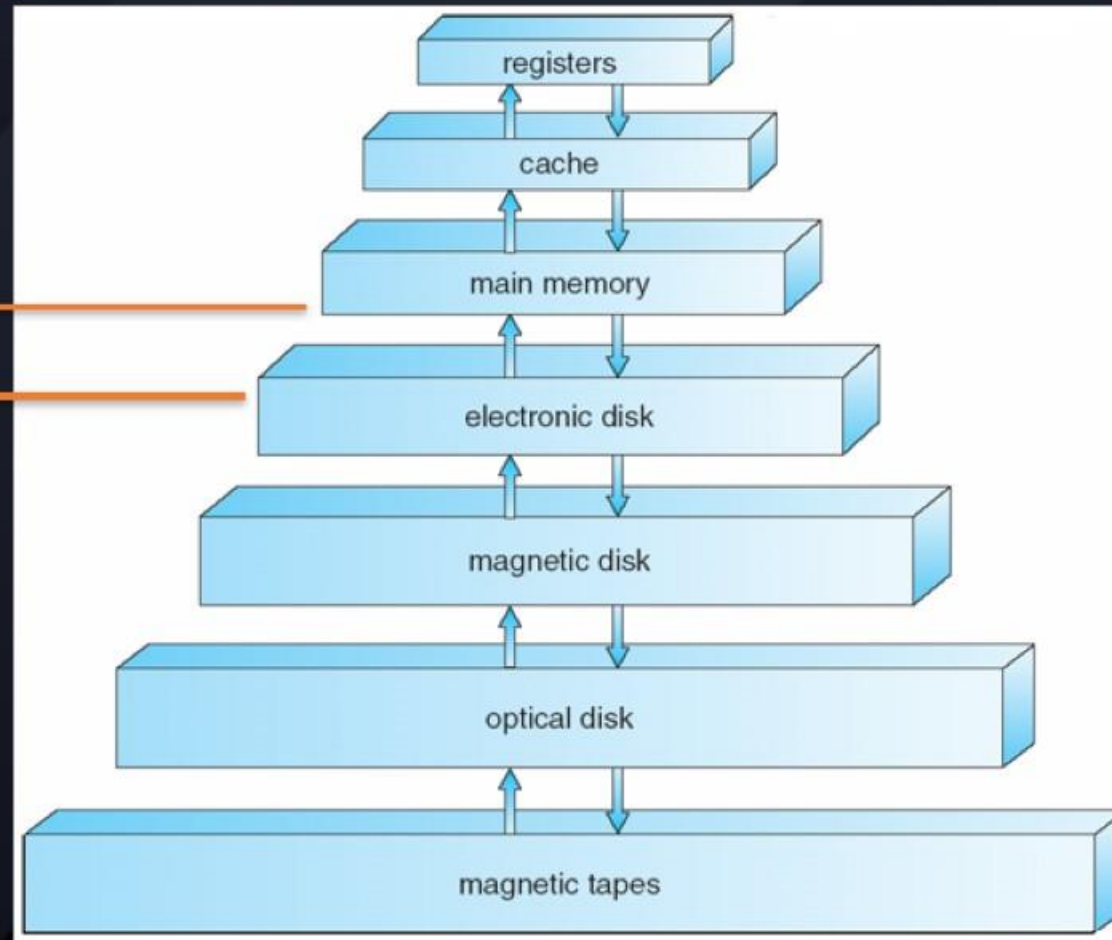
- Storage systems organized in hierarchy
 - Speed
 - Cost
 - Volatility
 - (Volatile: Loses its contents when power is removed)
 - (Non Volatile: Retains its contents even when power is removed)

Chapter 01

Storage Device Hierarchy

Smaller in size
But more costly(price)
Volatile
Faster (smaller Access time)

Larger in size
But less costly(price)
Non-Volatile
Slower (larger Access Time)



Programs are loaded into secondary storage for storage purposes
Programs are loaded into your main memory for execution purposes

- A magnetic disk is a storage device that uses a magnetization process to write, rewrite and access data.
- Electronic Disks: is data storage device functionally similar to a hard disk drive but using flash memory
- Cache memory: is a small-sized type of volatile computer memory that provides high-speed data access to a processor and stores frequently used computer programs, applications and data.

Basic of Operating Systems (I/O Structure)

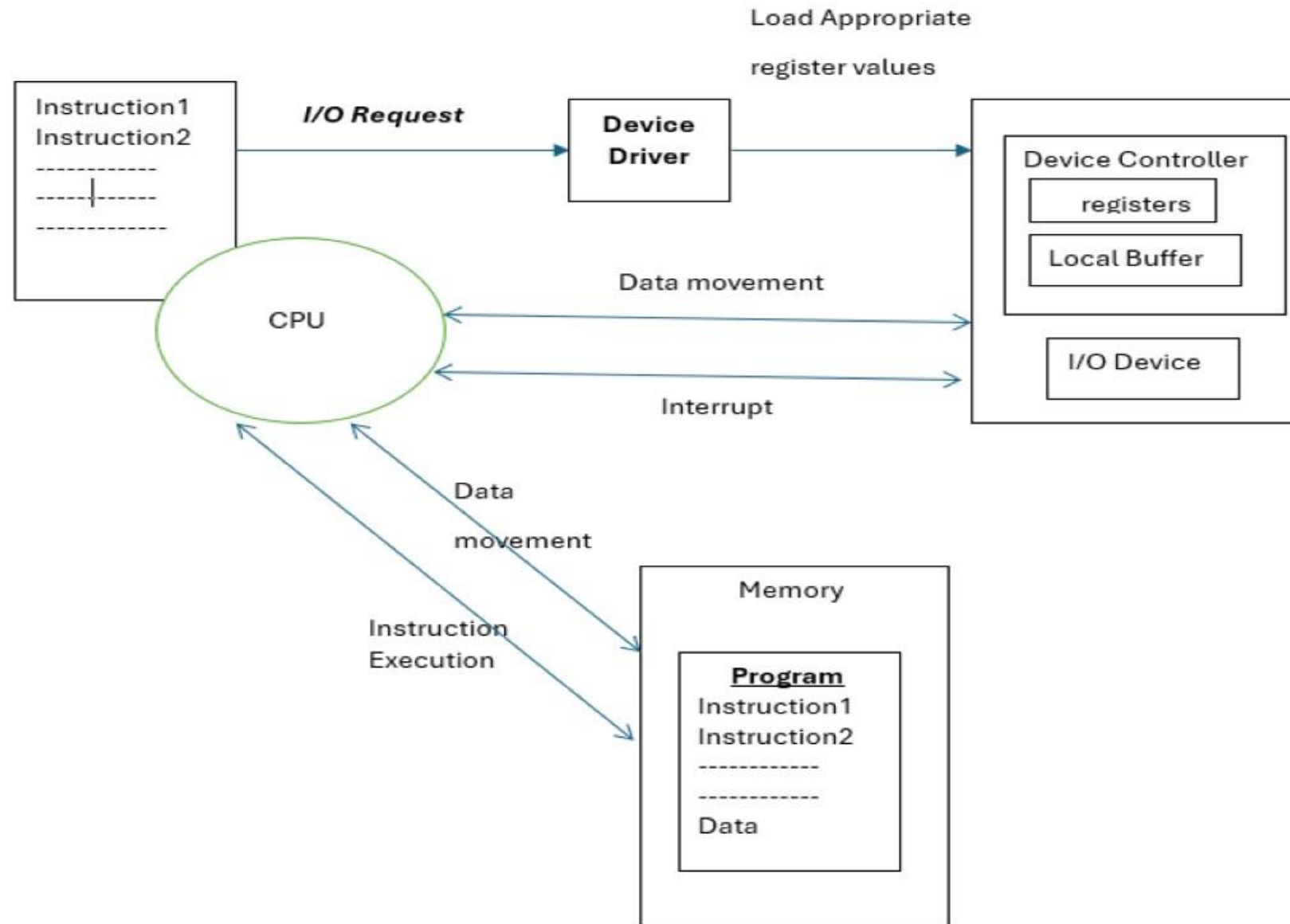
- Storage is only one of many types of I/O devices within a computer.
- A large portion of operating system code is dedicated to managing I/O, both because of its importance to the reliability and performance of system and because of the varying nature of the devices.
- A general-purpose computer system consists of CPUs and multiple device controllers that are connected through a common bus.

- Typically, Operating systems have a device driver for each device controller.
- This device driver understands the device controller and presents a uniform interface to the device to the rest of the operating system.

Working of an I/O operation

- To start in I/O operation (read from file for example) , the program, sends I/O request to the Operating system (the device driver), which in turn loads the appropriate registers within the device controller which in turn determine what action to take.
- The controller starts the transfer of data from the device to the local buffer.
- Once data transfer is completed, the device controller informs device driver via interrupt that it has finished its operation and the control is returned to the operating system

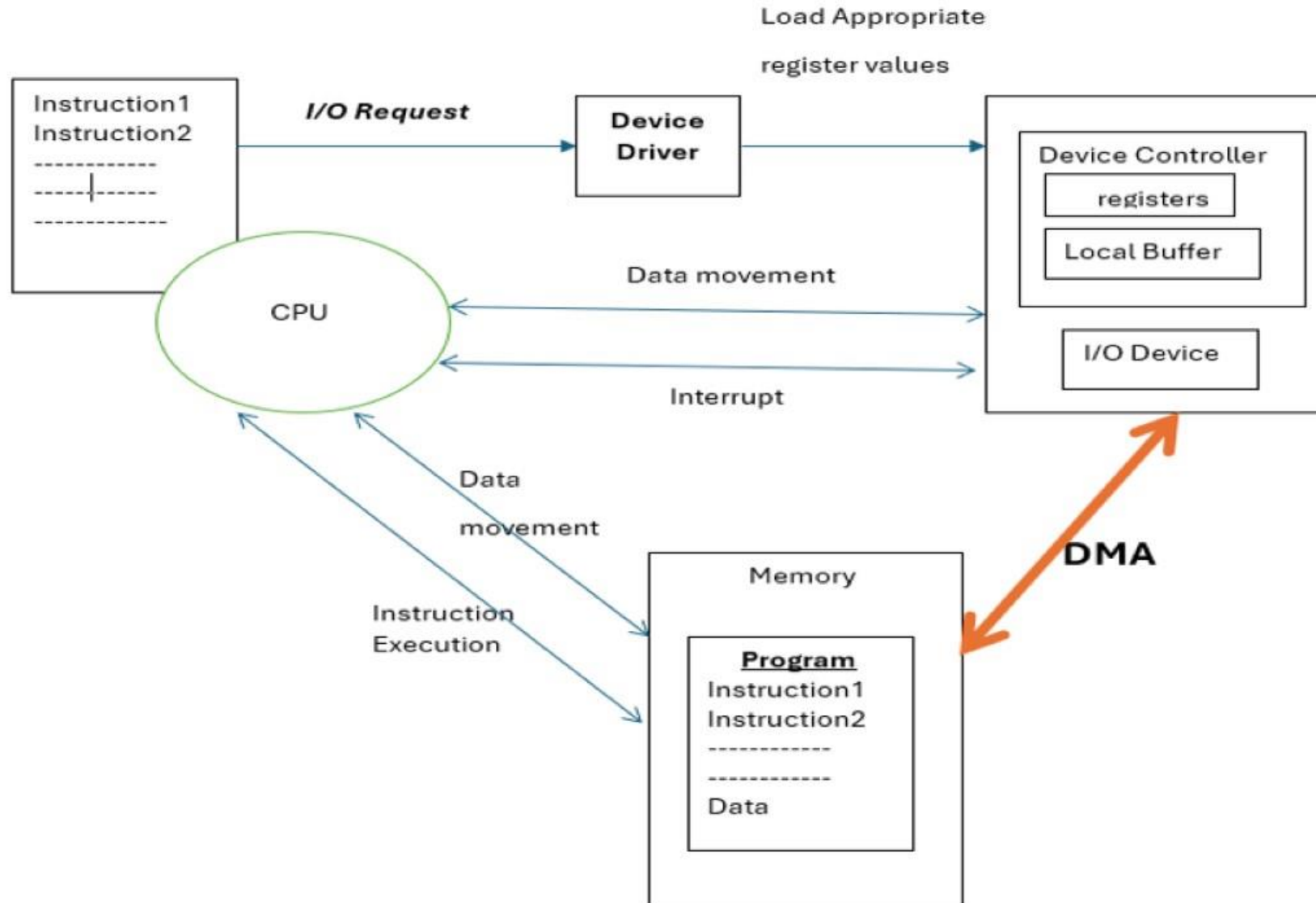
Chapter 01



Basic of Operating Systems (I/O Structure)

- The form presented in the last slide is called interrupt-driven I/O. it is suitable when moving small amounts of data but can produce high overhead when used for bulk of data.
- To solve this problem, Direct Memory Access (**DMA**) is used.
- Device controller with **DMA** capability, 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

Chapter 01



DMA is Used for high-speed I/O devices that is able to transmit information at close to memory speeds

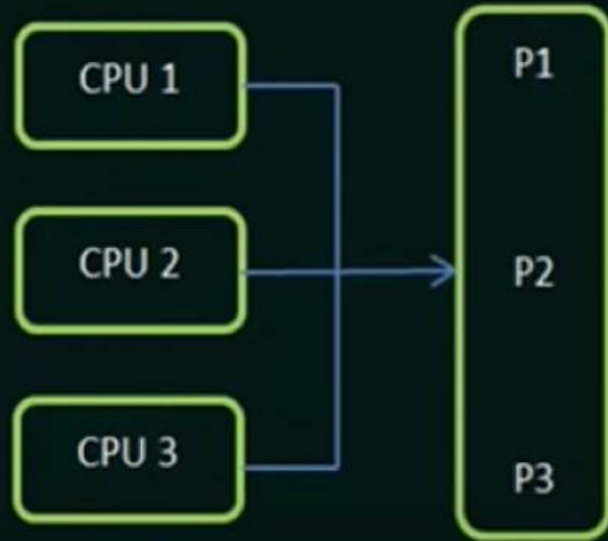
Computer-System Architecture

- Types of Computer Systems are based on the number of General-Purpose Processors:
 - Single Processor Systems
 - Multi Processor Systems
 - Clustered Systems
- Single Processor Systems: one main CPU capable of executing a general-purpose instruction set.
- Other special purpose processors (device controllers) are also present to perform specific task.

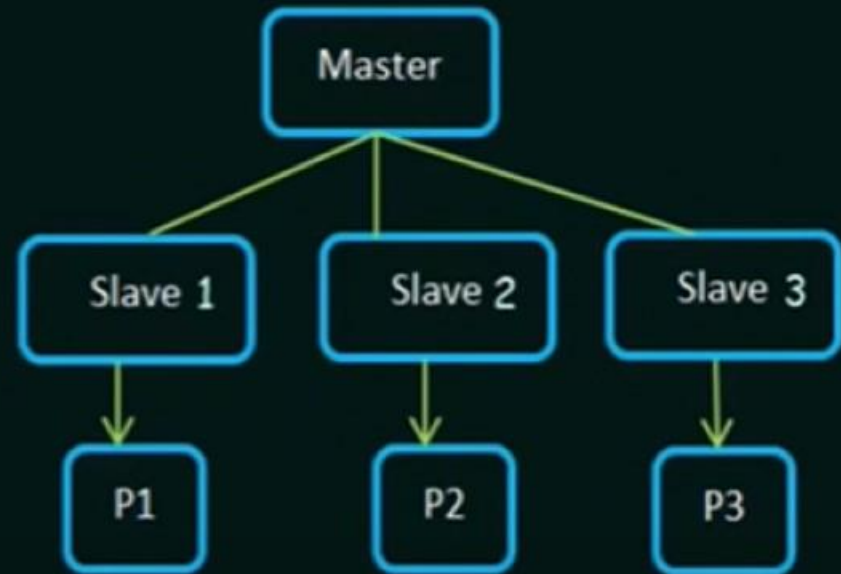
Computer-System Architecture

- Multiprocessors systems growing in use and importance
 - Also known as parallel systems, tightly-coupled systems
 - Advantages include
 - Increased throughput (amount of data transferred in unit of time)
 - Economy of scale
 - Increased reliability – graceful degradation or fault tolerance
 - Two types
 - Asymmetric Multiprocessing
 - Symmetric Multiprocessing

Symmetric Multiprocessing

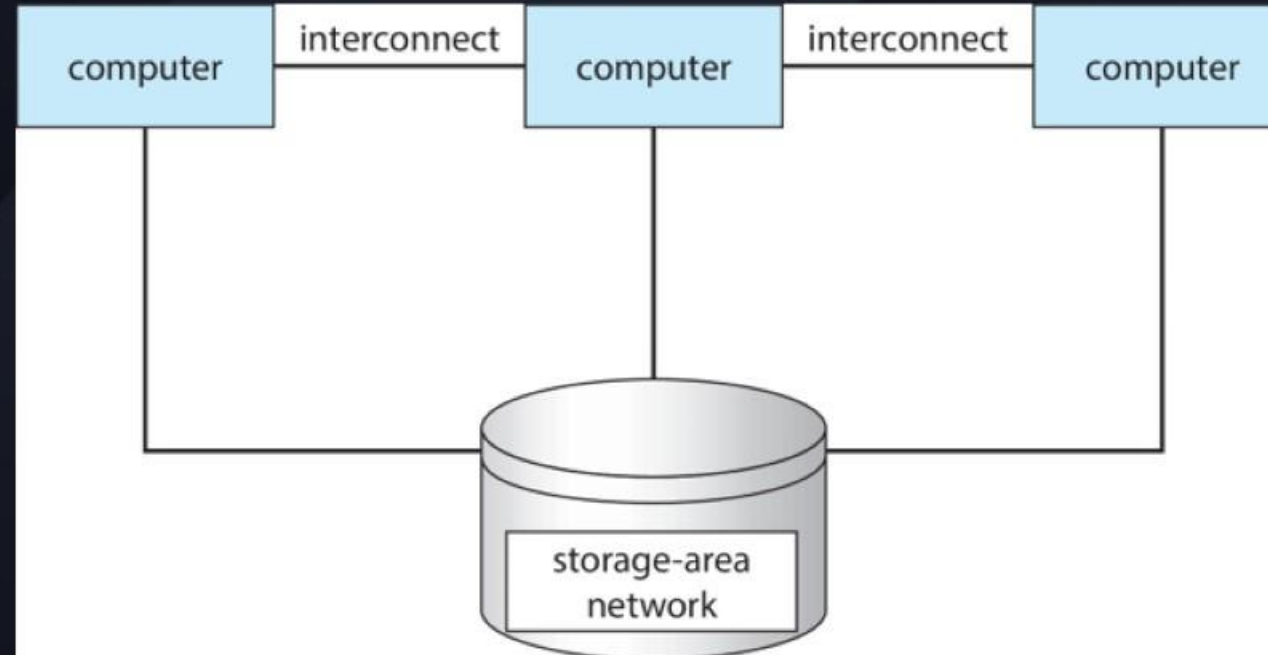


Asymmetric Multiprocessing



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



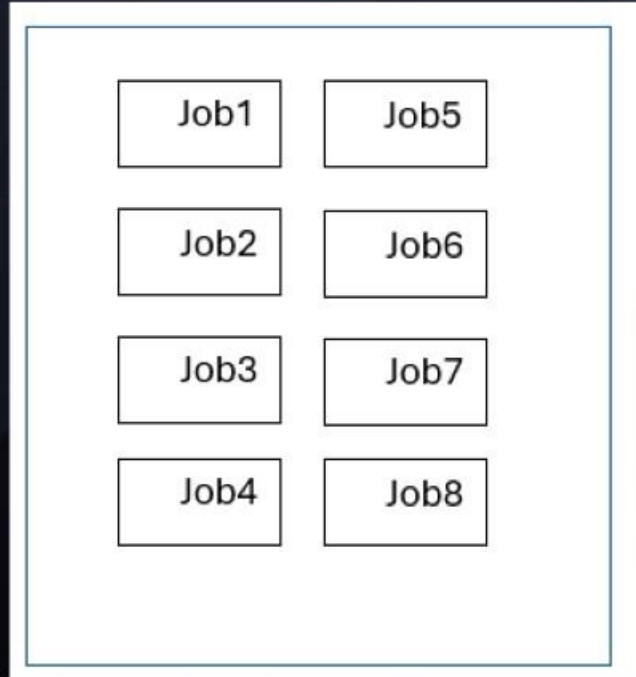
Operating Systems Structure

- Operating Systems vary greatly in their makeup internally:
- Multiprogramming
- Time Sharing (Multitasking)

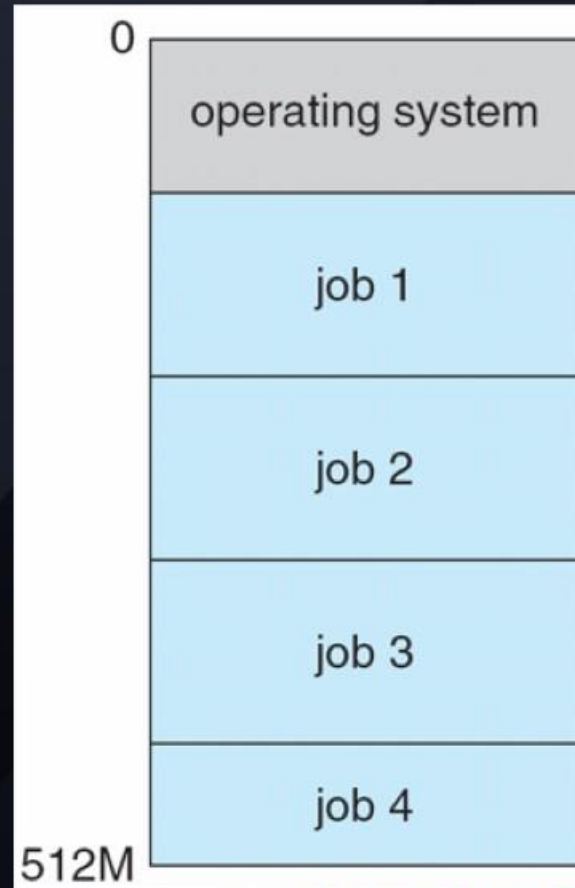
- **Multiprogramming** needed for efficiency
 - Single user cannot keep CPU and I/O devices busy at all times
 - Multiprogramming organizes jobs (**code and data**) so CPU always has one to execute
 - A subset of total jobs in system is kept in memory
 - One job selected and run via **job scheduling**
 - When it has to wait (for I/O for example), OS switches to another job
- In **Multiprogramming** provide efficient Resources utilization but not good in providing user interaction with the computer system

Chapter 01

Operating Systems Structure



Hard Disk



Memory layout of multiprogramming
systems



- **Timesharing (multitasking)** is a logical extension of multiprogramming in which 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 $\Rightarrow$ **process**

Operating Systems Structure

- If several jobs ready to run at the same time $\Rightarrow$ **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 Systems Operations

- Operating systems are considered as an Interrupt driven mostly by hardware
- A *trap* is a software-generated interrupt caused either by an error or a user request
- Software error or request creates **exception** or **trap**
 - Division by zero, request for operating system service

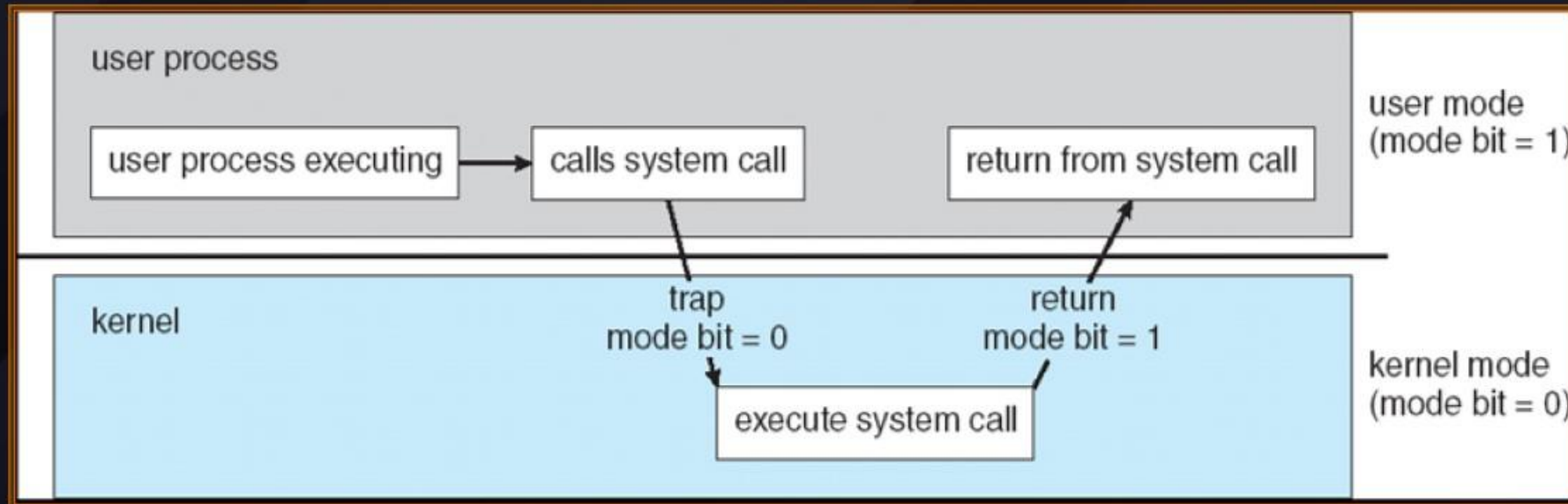
Operating Systems Operations

- **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
 - Set interrupt after specific period
 - Operating system decrements counter
 - When counter zero generate an interrupt
 - Set up before scheduling process to regain control or terminate program that exceeds allotted time

Transition from User to Kernel Mode



Transition from User to Kernel Mode

- User's code issues a system call
- Mode is changed to kernel mode
- OS first checks that everything (e.g., parameter values) is in order and legal
- Then, OS executes system call which may contain multiple privileged instructions
- mode is set again to user mode

End of Chapter 1