

Operating Systems

Prepared by Dr. Alaa Mohamed

Lecture 1

جامعة الجميع الذكية

تسعى لتكون الكيان الأول الذي يتبنى مبدأ المجتمع التعليمي
التعاوني ويقدم العلم مقابل العمل لتنمية وبناء المجتمعات.

Operating System Introduction



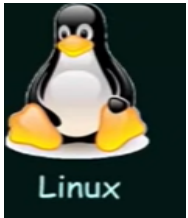
جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

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

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

Chapter 01

What is an Operating System?



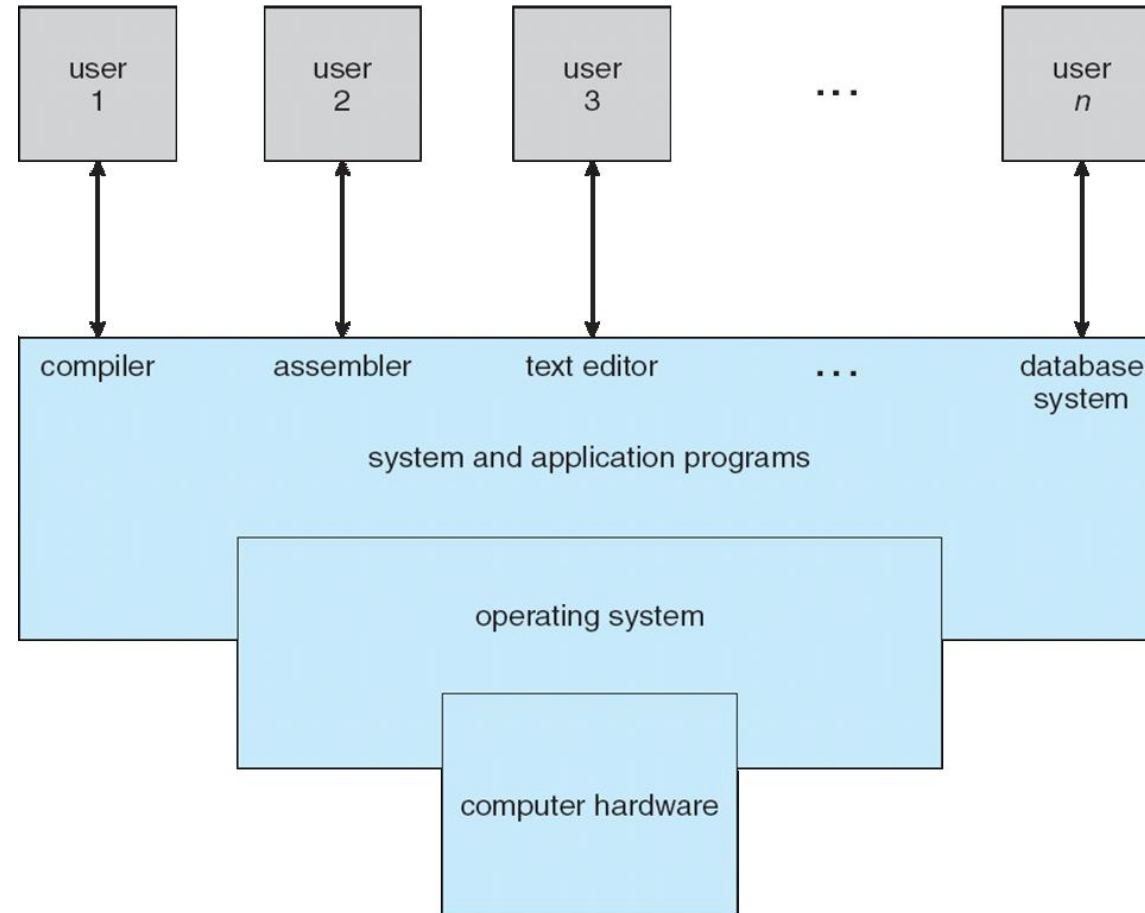
- A program that acts as an intermediary between a user of a computer and the computer hardware
- Operating system goals:
 - Execute user programs and make solving user problems easier
 - Make the computer system convenient to use
 - Use the computer hardware in an efficient manner

- Computer system 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

Chapter 01

Four Components of a Computer System



- 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

- “The one program running at all times on the computer” is the **kernel**. Everything else is either a system program (ships with the operating system) or an application program

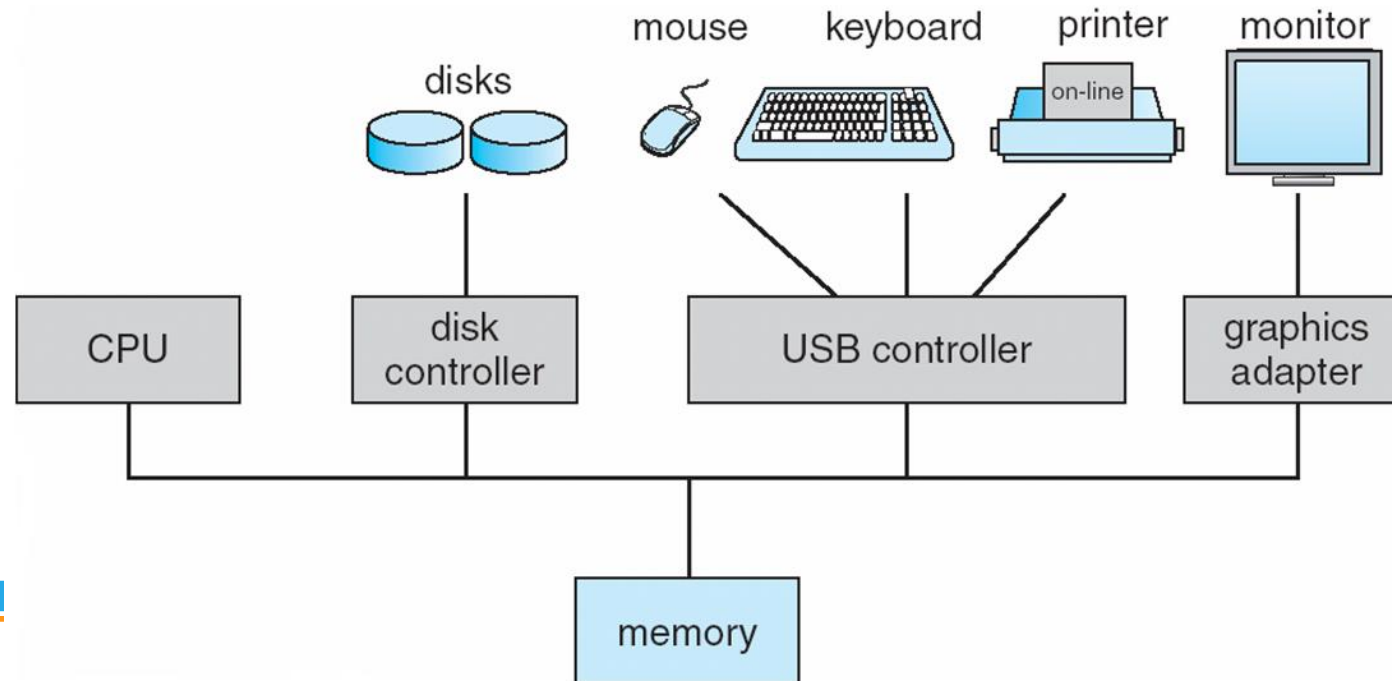
- **Bootstrap Program:-**

- The Initial program that runs when the Computer is powered up or rebooted
- It is stored in the ROM.
- It must know how to load the OS and start executing the System
- It must locate and Load into memory the OS kernel.

Chapter 01

Computer System Organization

- Computer-system operation
 - 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



- 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*

- **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

- 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

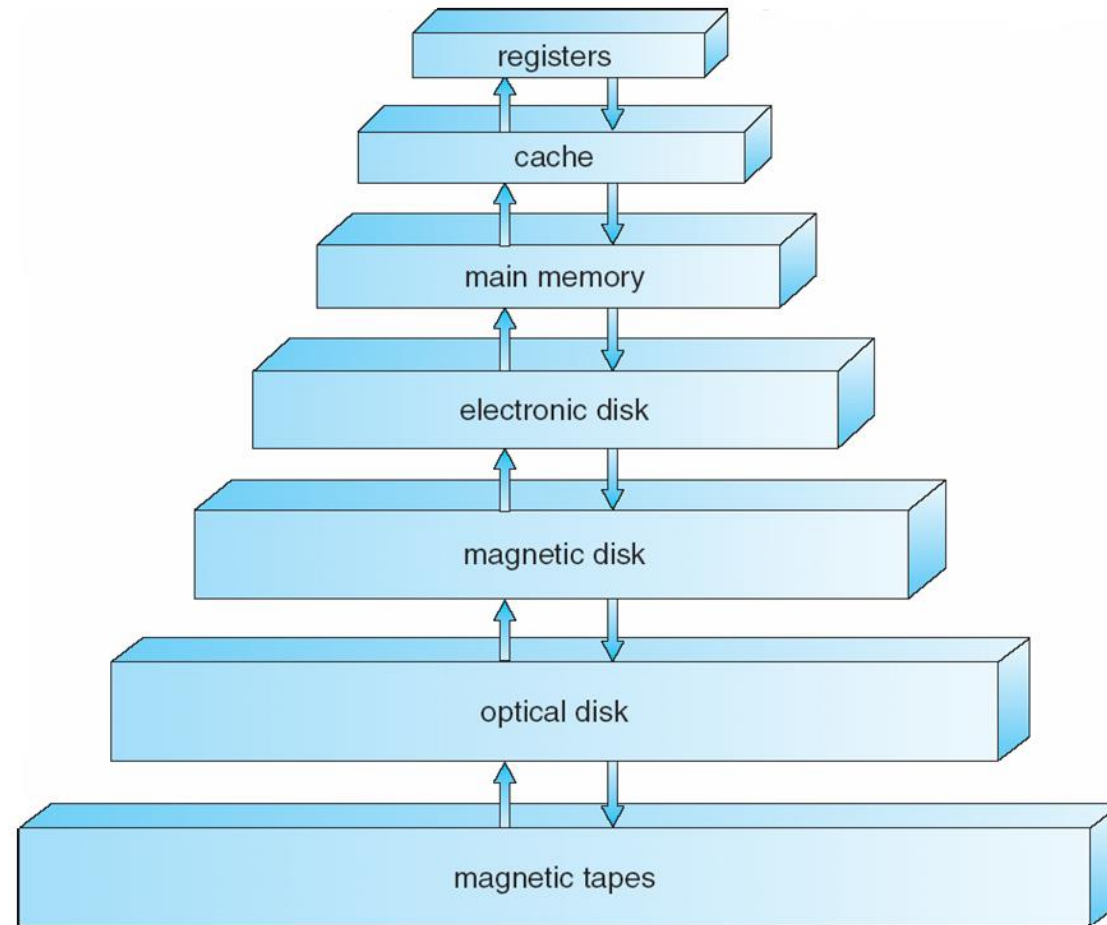


- A magnetic disk is a storage device that uses a magnetization process to write, rewrite and access data.
- Electronic Disks: is data storage device functionally similar to a hard disk drive but using flash memory
- Cache memory: is a small-sized type of volatile computer memory that provides high-speed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

- 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

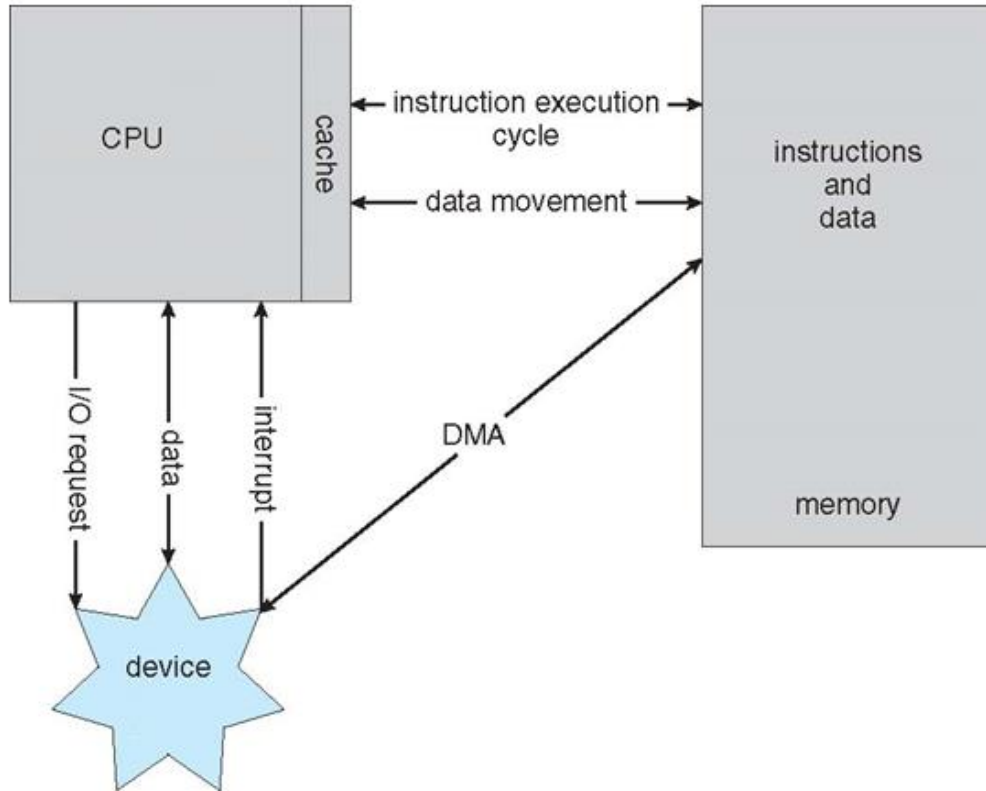


- Storage is only one of many types of I/O devices within a computer.
- A large portion of operating system code is dedicated to managing I/O, both because of its importance to the reliability and performance of system and because of the varying nature pf the devices.
- A general-purpose computer system 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.

Chapter 01

Working of an I/O operation

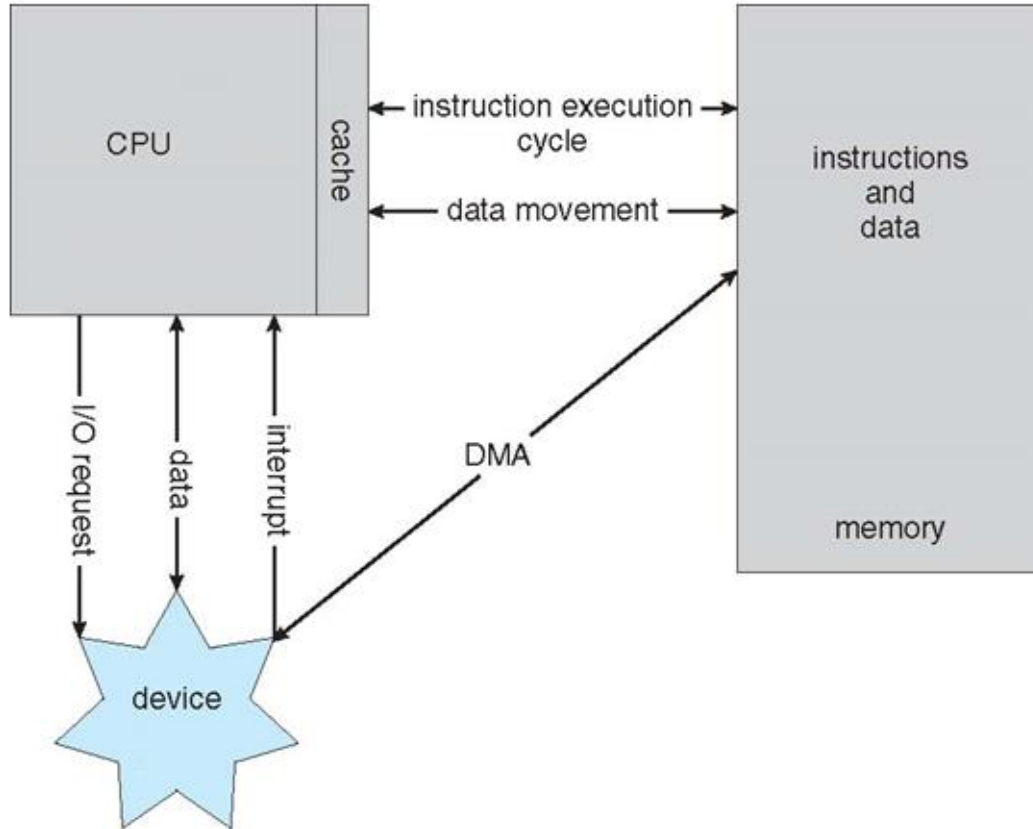


- To start in I/O operation, the device driver loads the appropriate registers within the device controller which in turn determine what action to take.
- The controller starts the transfer of data from the device to the local buffer.



Chapter 01

Working of an I/O operation



- 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

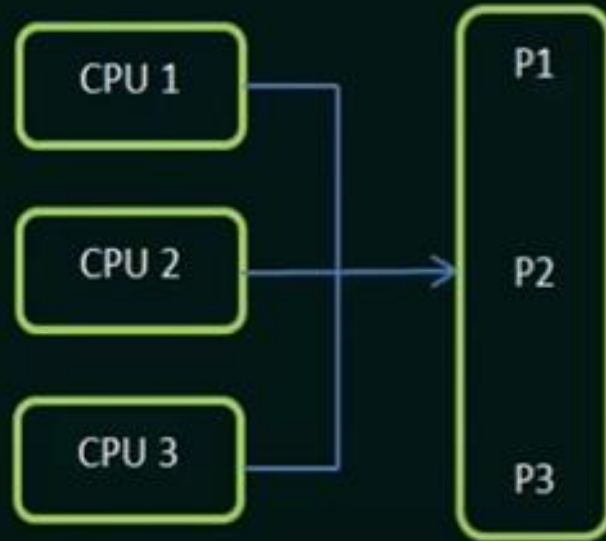


- The form presented in the last slide is called interrupt-driven I/O. it is suitable when moving small amounts of data but can produce high overhead when used for bulk of data.
- To solve this problem, Direct Memory Access (**DMA**) is used.
- Device controller with **DMA** capability, transfers blocks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

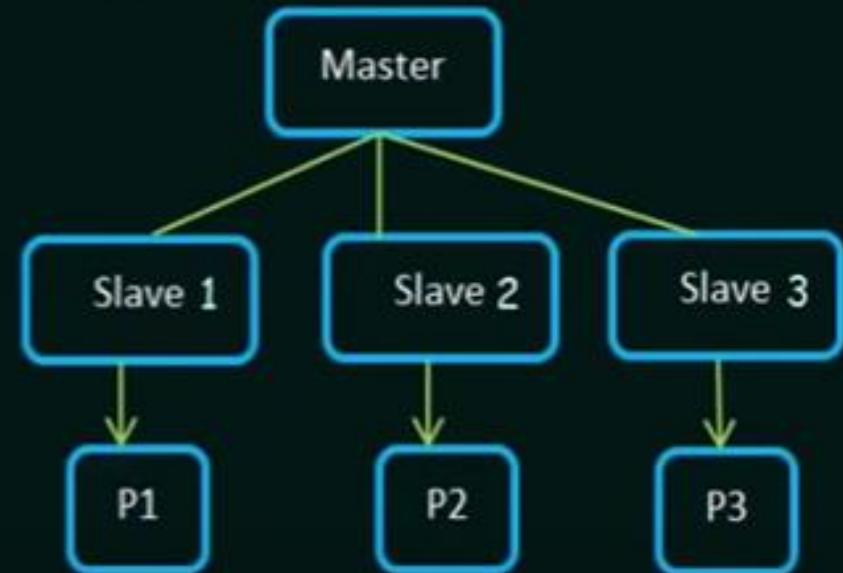
- Types of Computer Systems are based on the number of General-Purpose Processors:
 - Single Processor Systems
 - Multi Processor Systems
 - Clustered Systems
- **Single Processor Systems:** one main CPU capable of executing a general-purpose instruction set.
- Other special purpose processors (device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

- **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
 2. 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)**

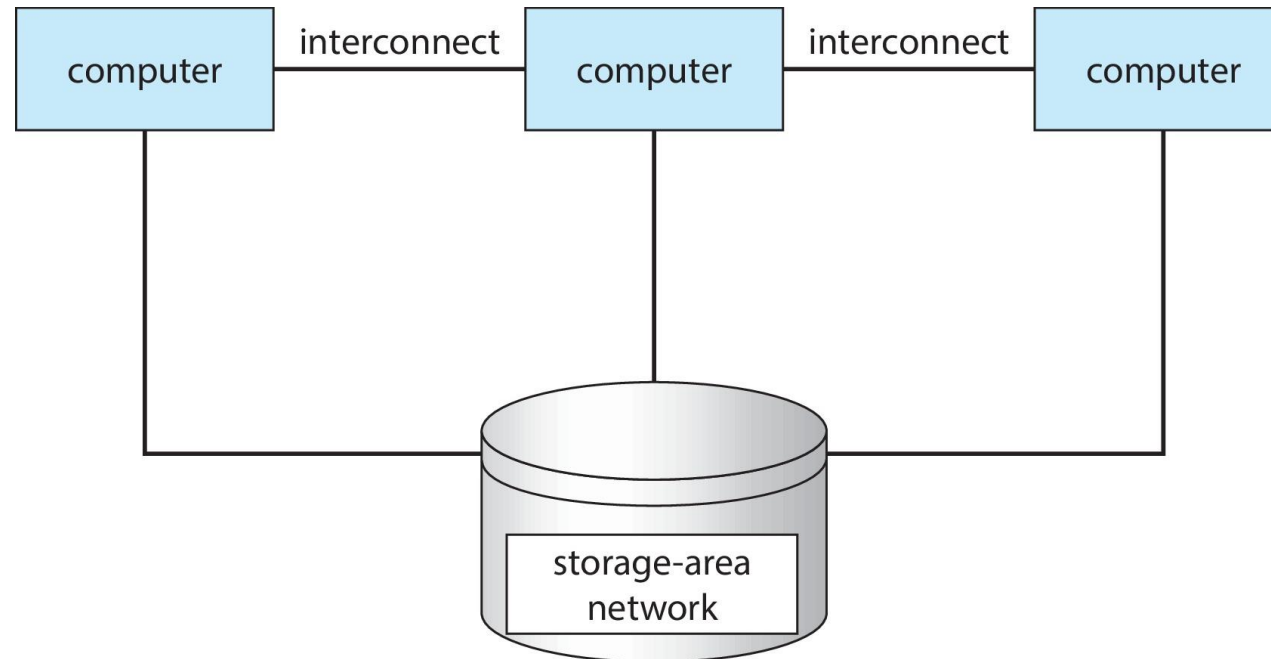
Provides a **high-availability** service which survives failures

Asymmetric clustering has one machine in hot-standby mode

Symmetric clustering has multiple nodes running applications, monitoring each other

Chapter 01

Computer-System Architecture



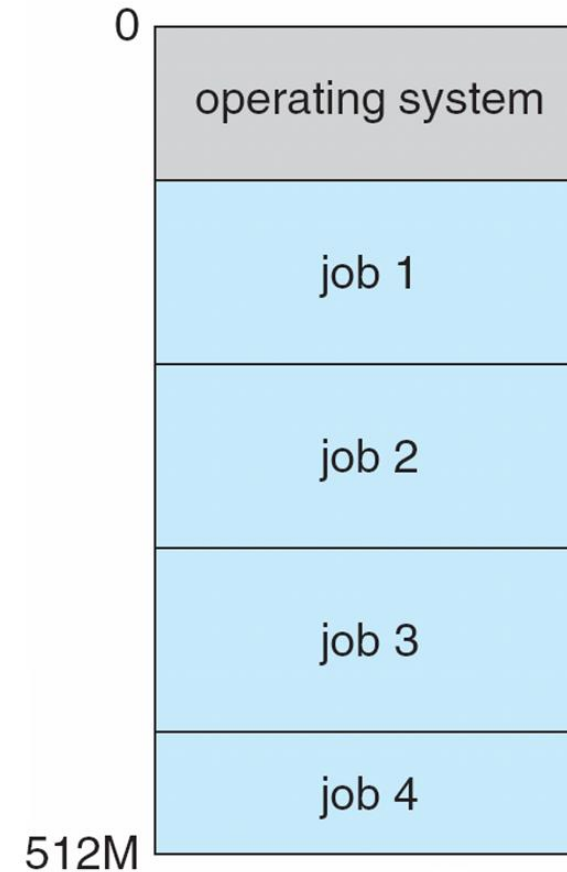
Operating Systems vary greatly in their makeup internally:

- 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

Chapter 01

Operating Systems Structure



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**

- 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

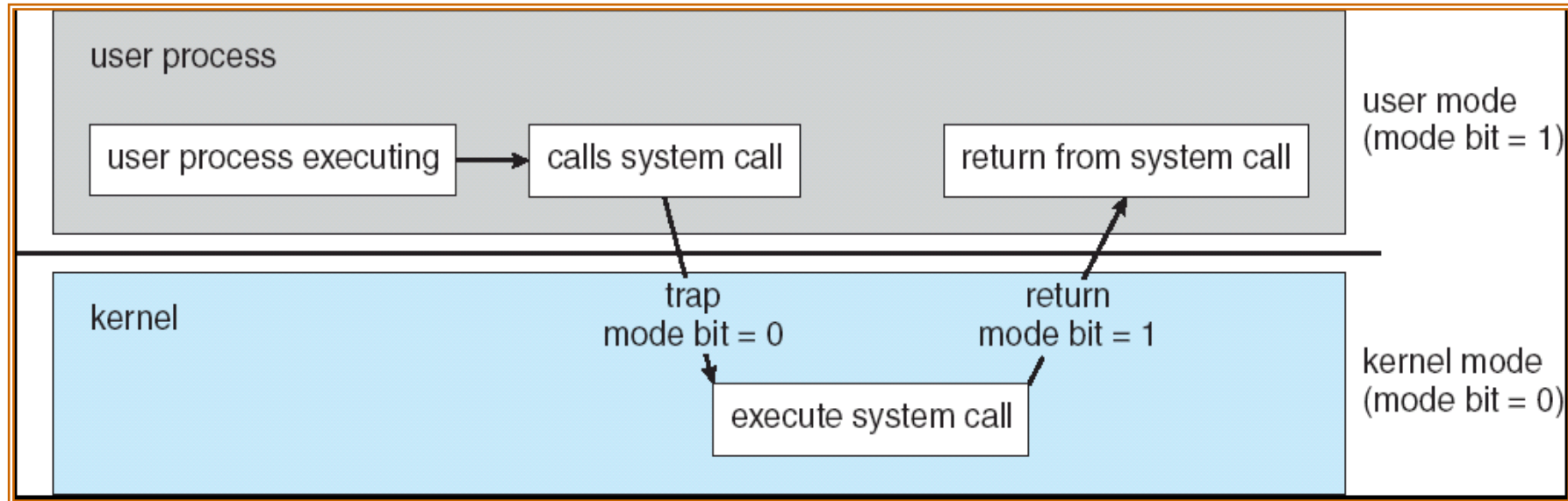
- Operating systems are considered as an Interrupt driven mostly by hardware
- A *trap* is a software-generated interrupt caused either by an error or a user request
- Software error or request creates **exception** or **trap**
 - Division by zero, request for operating system service

- **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

- 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

Chapter 01

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

