CHAPTER - 1

Introduction & Basics

Neso Academy

Introduction to OPERATING SYSTEM

Windows

Linux

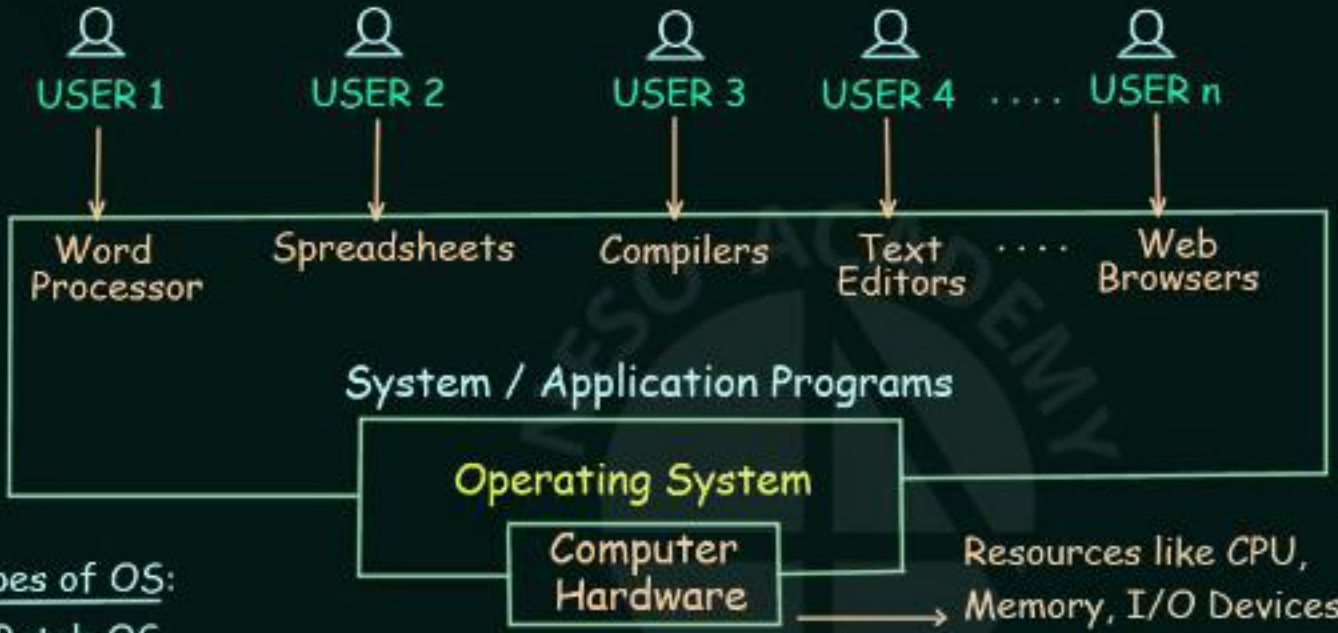
Ubuntu

Mac OS X
iOS

Android

→ An Operating System (OS) is a program that manages the computer hardware.

→ It also provides a basis for Application Programs and acts as an intermediary between computer User and compter Hardware.



Resources like CPU, Memory, I/O Devices

Types of OS:

- → Batch OS
- → Time sharing OS
- → Distributed OS
- → Network OS
- → Real Time OS
- → Multi Programming/ Processing/ Tasking OS

Goals of OS:

i) Convenience
ii) Efficiency
iii) Both

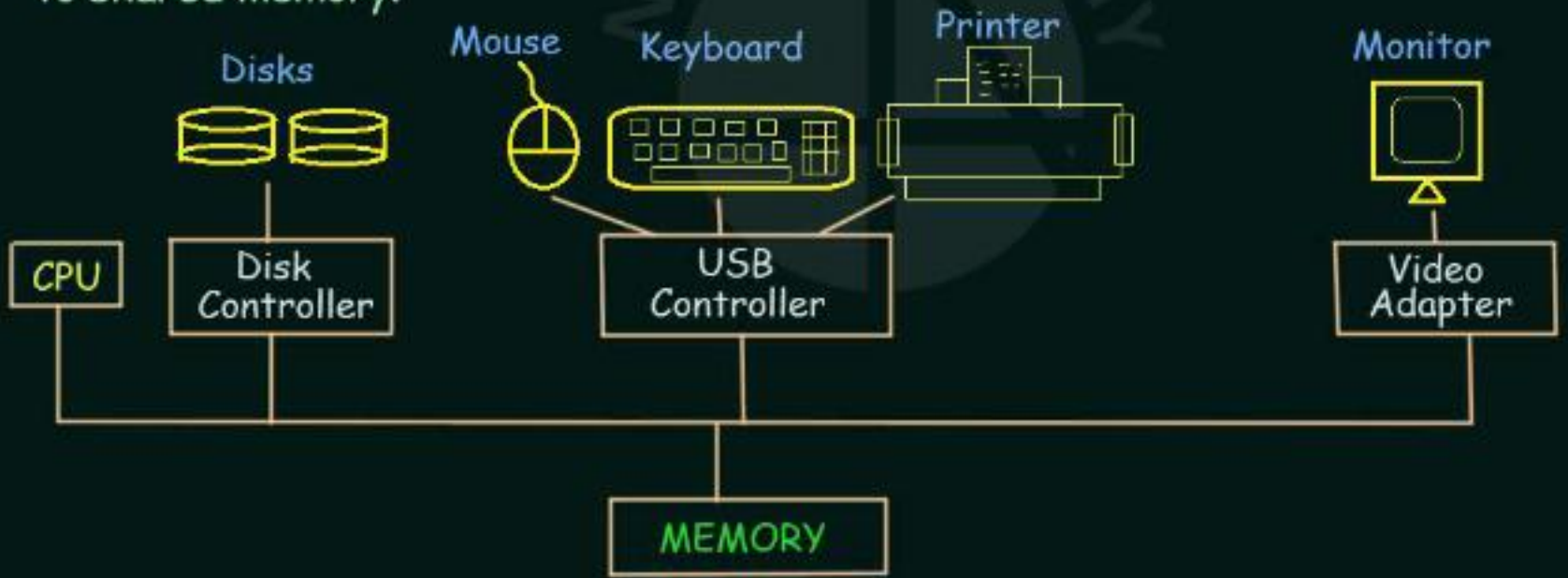
Functions of OS:

- → It is an interface between User & Hardware
- → Allocation of Resources
- → Management of Memory, Security, etc.

Basics of Operating System (Computer System Operation)

Some basic knowledge of the structure of Computer System is required to understand how Operating Systems work.

→ 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

→ The CPU and the device controllers can execute concurrently, competing for memory cycles

→ To ensure orderly access to the shared memory, a memory controller is provided whose function is to synchronize access to the memory

Some important terms:

1) Bootstrap Program: → The initial program that runs when a computer is powered up or rebooted.
→ It is stored in the ROM.
→ It must know how to load the OS and start executing that system.
→ It must locate and load into memory the OS Kernel.

2) Interrupt: → The occurence of an event is usually signalled by an 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 the system bus.

3) System Call (Monitor call): → Software may trigger an interrupt by executing a special operation called System Call.

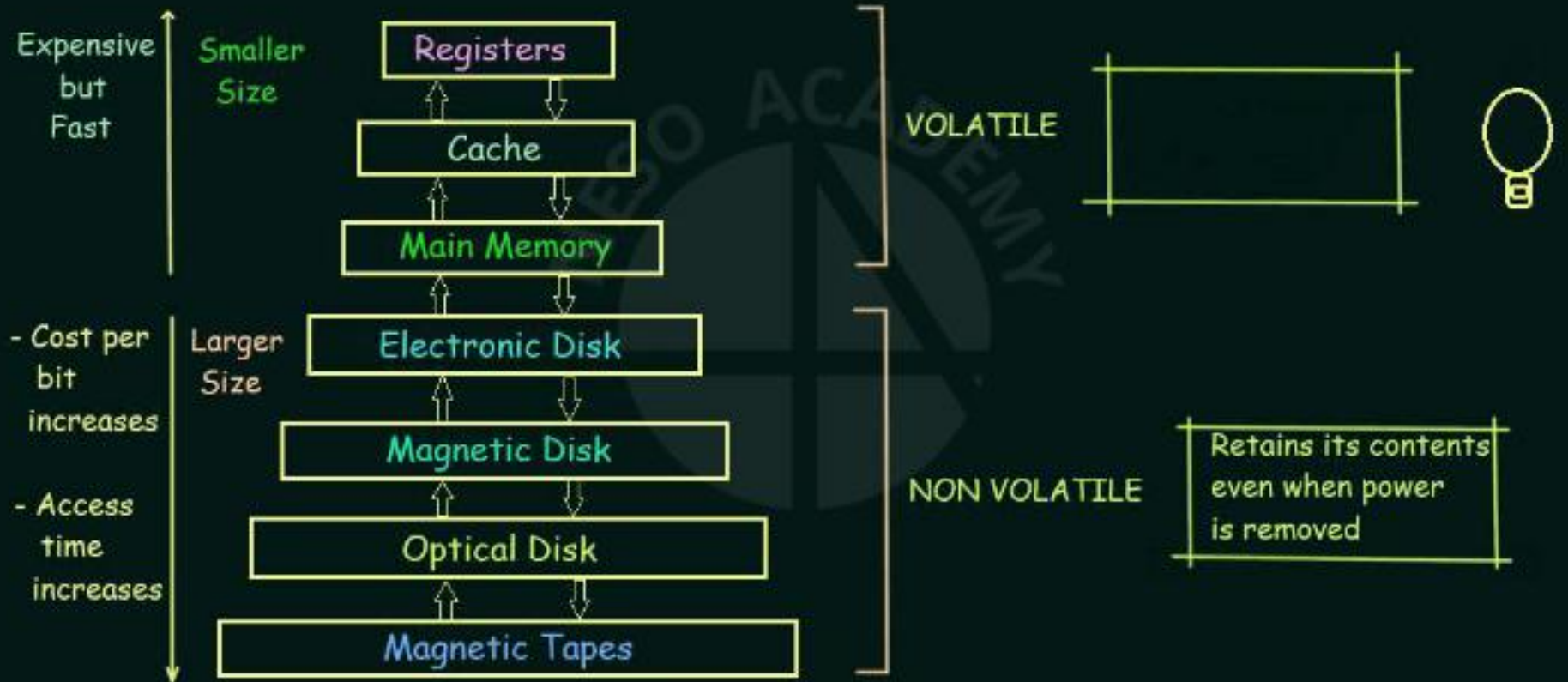
When the CPU is interrupted, it stops what it is doing and immediately transfers execution to a fixed location.

→ The fixed location usually contains the starting address where the Service Routine of the interrupt is located.

The Interrupt Service Routine executes.

On completion, the CPU resumes the interrupted computation.

Basics of Operating System (Storage Structure)



Basics of Operating System (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 a system and because of the varying nature of the devices

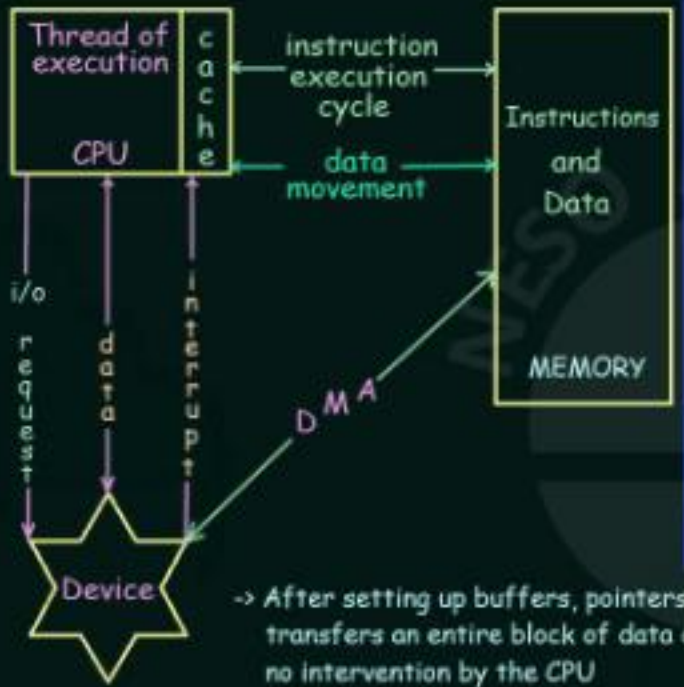
-> A general-purpose computer system consists of CPUs and multiple device controllers that are connected through a common bus

-> Each device controller is in charge of a specific type of device

-> 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 an I/O operation, the device driver loads the appropriate registers within the device controller

-> The device controller, in turn, examines the contents of these registers to determine what action to take

-> The controller starts the transfer of data from the device to its local buffer

-> Once the transfer of data is complete, the device controller informs the device driver via an interrupt that it has finished its operation

-> The device driver then returns control to the operating system

This form of interrupt-driven I/O is fine for moving small amounts of data but can produce high overhead when used for bulk data movement

To solve this problem, Direct Memory Access (DMA) is used

-> After setting up buffers, pointers, and counters for the I/O device, the device controller transfers an entire block of data directly to or from its own buffer storage to memory, with no intervention by the CPU

-> Only one interrupt is generated per block, to tell the device driver that the operation has completed

-> While the device controller is performing these operations, the CPU is available to accomplish other works

Computer System Architecture

Types of Computer Systems based on number of General Purpose Processors:

1. Single Processor Systems 

2. Multiprocessor Systems 

3. Clustered Systems 

1. Single Processor Systems 

- Has two or more processors in close communication, sharing the computer bus and sometimes the clock, memory, and peripheral devices

Advantages:

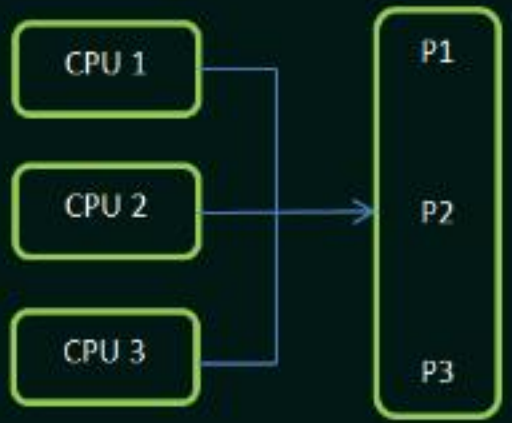
👍 Increased throughput

👍👍 Economy of scale

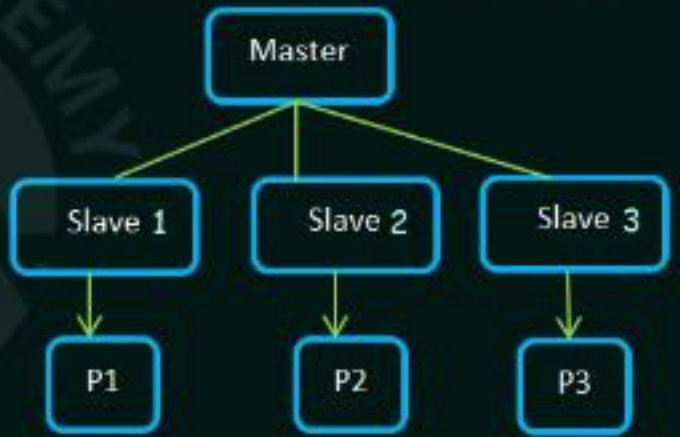
👍👍👍 Increased reliability

Types of Multiprocessor Systems:

Symmetric Multiprocessing



Asymmetric Multiprocessing



3. Clustered Systems

- Like multiprocessor systems, clustered systems gather together multiple CPUs to accomplish computational work.
- They are composed of two or more individual systems coupled together.
- Provides high availability
- Can be structured asymmetrically or symmetrically

asymmetrically

- One machine in Hot-Standby mode
- Others run applications

symmetrically

- Two or more hosts run applications
- Monitors each other

www.nesoacademy.org

Neso Academy