OPERATING SYSTEMS

Computer System Organisation

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

OPERATING SYSTEMS

Course Syllabus - Unit 1

UNIT 1: Introduction and Process Management

Operating-System Structure & Operations, Kernel Data Structures,Computing Environments, Operating-System Services, OperatingSystem Design and Implementation. Process concept: Process in memory, Process State, Process Control Block, Process Creation and Termination, CPU Scheduling and Scheduling Algorithms, IPC - Shared Memory & Message Passing, Pipes - Named and Ordinary. Case Study: Linux/Windows Scheduling Policies.

OPERATING SYSTEMS

Course Outline

Class No.	Chapter Title / Reference Literature	Topics to be covered	% of Portions covered	
			Reference chapter	Cumulative
1	1.1-1.2	What Operating Systems Do, Computer-System Organization?	1	21.4
2	1.3,1.4,1.5	Computer-System Architecture, Operating-System Structure & Operations	1	
3	1.10,1.11	Kernel Data Structures, Computing Environments	1	
4	2.1,2.6	Operating-System Services, Operating System Design and Implementation	2	
5	3.1-3.3	Process concept: Process in memory, Process State, Process Control Block, Process Creation and Termination	3	
6	5.1-5.2	CPU Scheduling: Basic Concepts, Scheduling Criteria	5	
7	5.3	Scheduling Algorithms: First-Come, First-Served Scheduling, Shortest-Job-First Scheduling	5	
8	5.3	Scheduling Algorithms: Shortest-Job-First Scheduling (Pre-emptive), Priority Scheduling	5	
9	5.3	Round-Robin Scheduling, Multi-level Queue, Multi-Level Feedback Queue Scheduling	5	
10	5.5,5.6	Multiple-Processor Scheduling, Real-Time CPU Scheduling	5	
11	5.7	Case Study: Linux/Windows Scheduling Policies	5	
12	3.4,3.6.3	IPC – Shared Memory & Message Passing, Pipes – Named and Ordinary	3,6	

Topic Outline

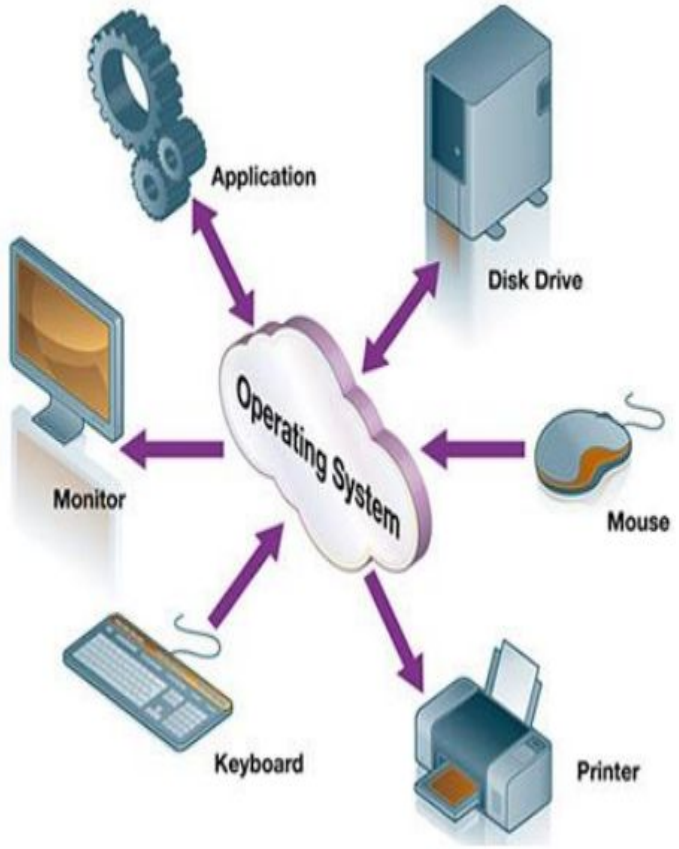
- Need for an Operating System
- Operating System Goals
- Why Study Operating Systems ?
- Computer System Structure
- What Operating Systems Do ?
- Computer System Organization

Topic Outline

- Computer System Operation
- Common Functions of Interrupts
- Interrupt Handling
- Interrupt Timeline for a single process doing output
- Program Execution Model
- Storage Structure
- Typical Q and As

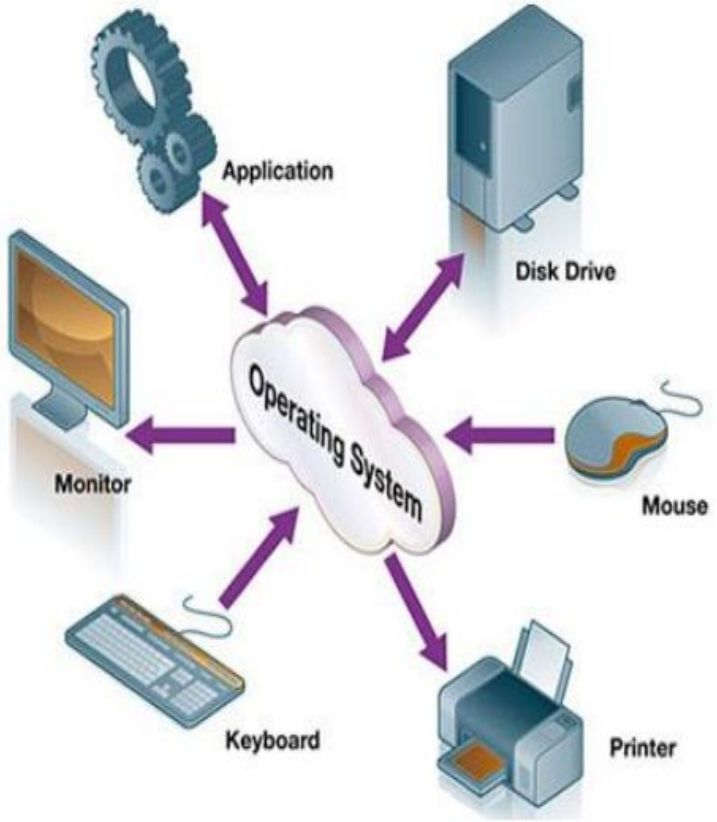
Need for an Operating System

- If **Operating System**, is not developed then how will the application developers access the hardware ?



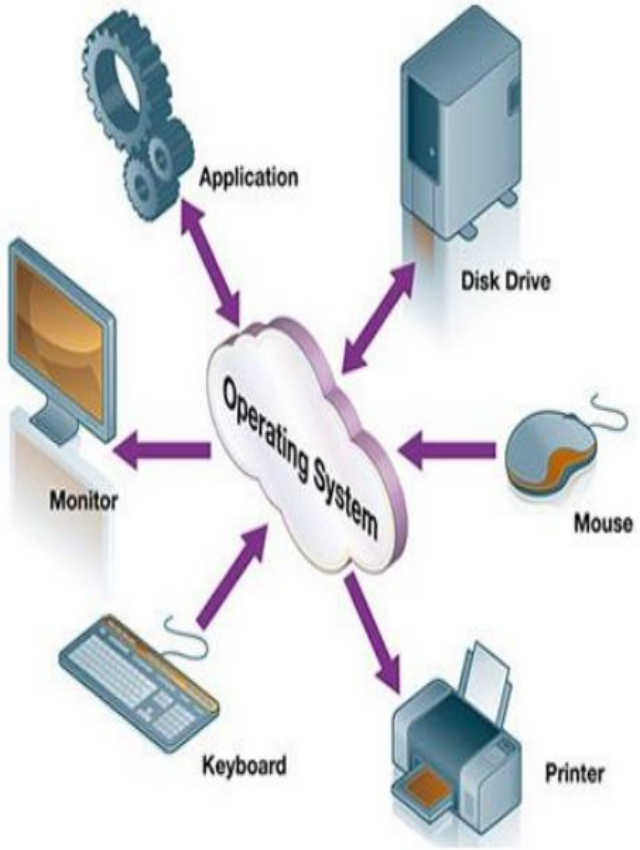
Operating System Goals

- Execute user programs and make solving user problems easier
- Make the computer system convenient to use
- Use the computer hardware in an efficient manner
- Manage resources such as
  - Memory
  - Processor(s)
  - I/O Devices



Why Study Operating Systems ?

- **Since only a small percentage of computer practitioners will be involved in the creation or modification of an operating system. Why is that everyone should study OS as core course in CS ?**
  - **Almost all code runs on top of an Operating System, and thus knowledge of how operating systems work is crucial to proper, efficient, effective, and secure programming.**
    - **This can be achieved by understanding the fundamental concepts of operating systems**



Computer System Structure

- **Hardware** – provides basic computing resources such as 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

What Operating Systems Do ?

- Depends on the point of User View and System View

User's View

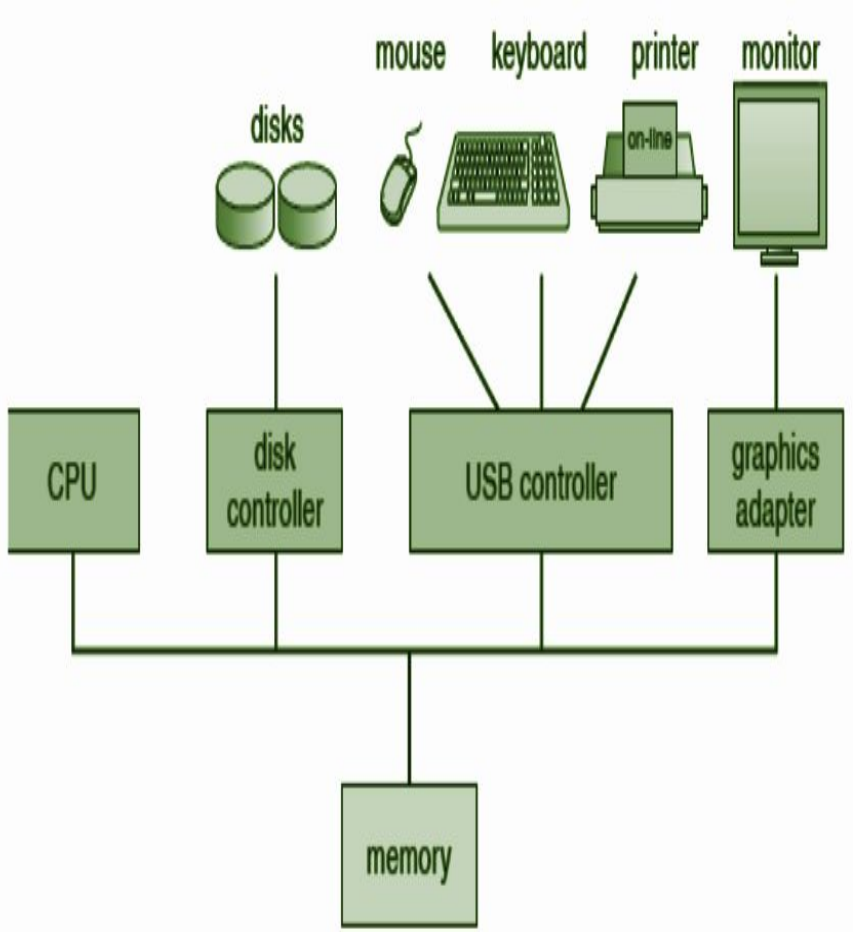
- Users want convenience, ease of use and good performance
- Users don't care about resource utilization
- But shared computer such as mainframe or minicomputer must keep all users happy.
  - **Maximize** resource utilization.
  - Available CPU time, memory, and I/O are used efficiently and that no individual user takes more than allotted / eligible **fair share**

System View

- 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

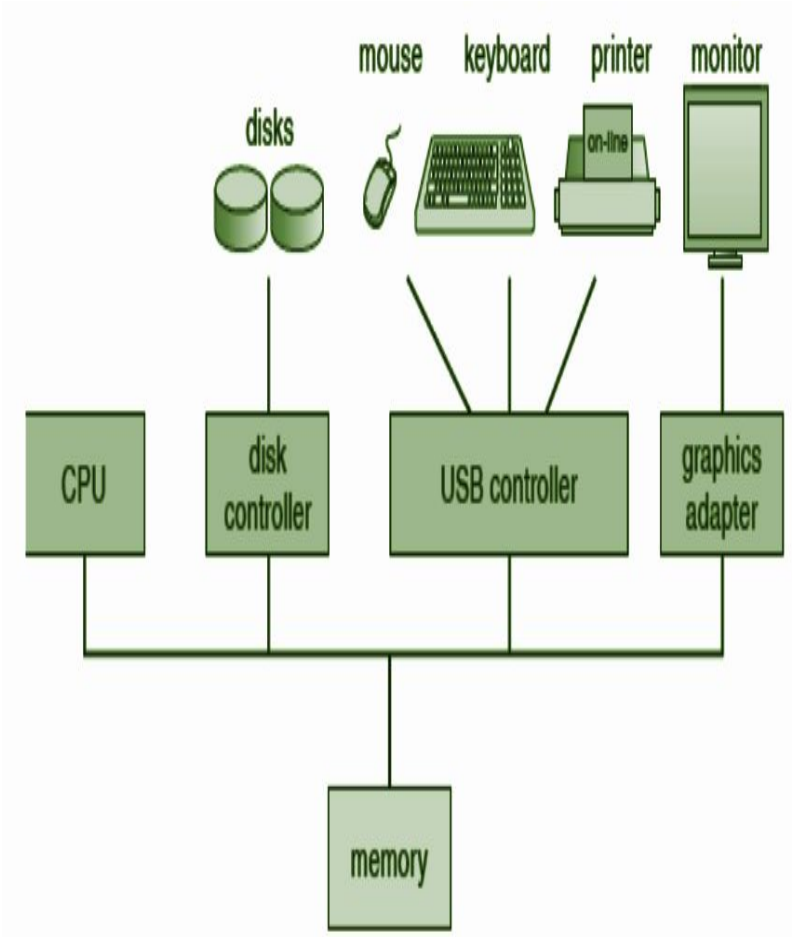
Computer System Organization

- Computer-system consists of
  - One or more **CPUs**, device controllers connect through common bus providing access to shared memory
  - The CPU and the **device controllers** can execute **concurrently**, competing for memory cycles.
  - **Memory controller** is provided to synchronize access to the memory.



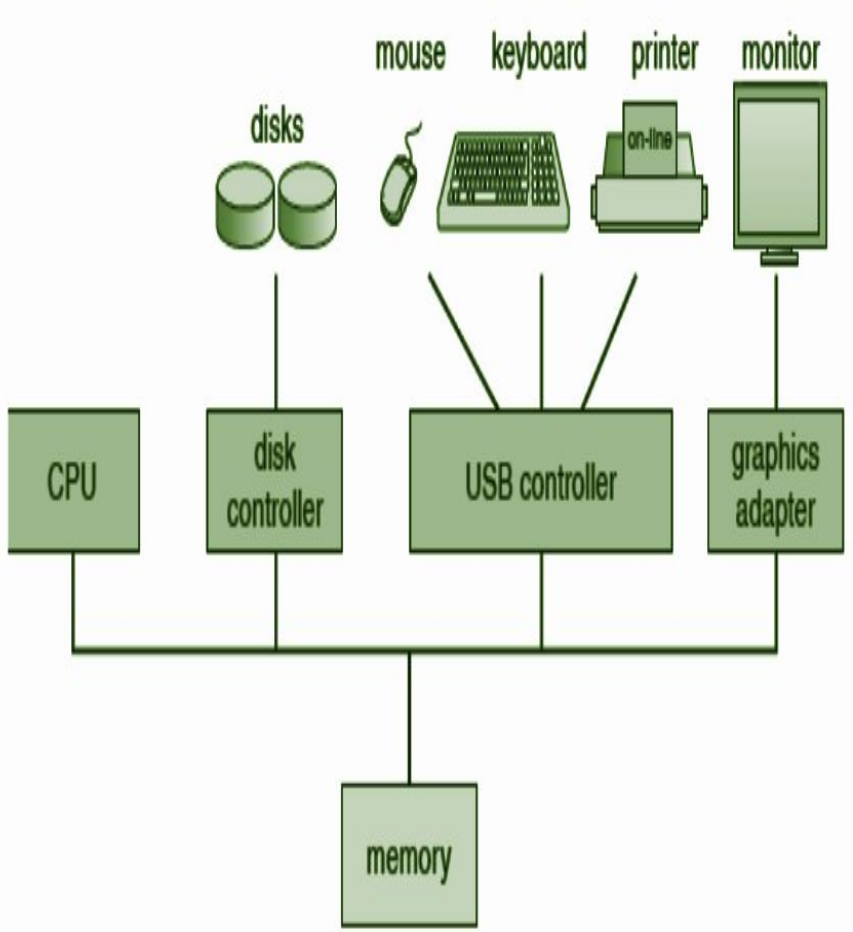
Computer System Operation

- **Input/Output 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**
- Each device controller has registers for action (like “read character from keyboard”) to take 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**



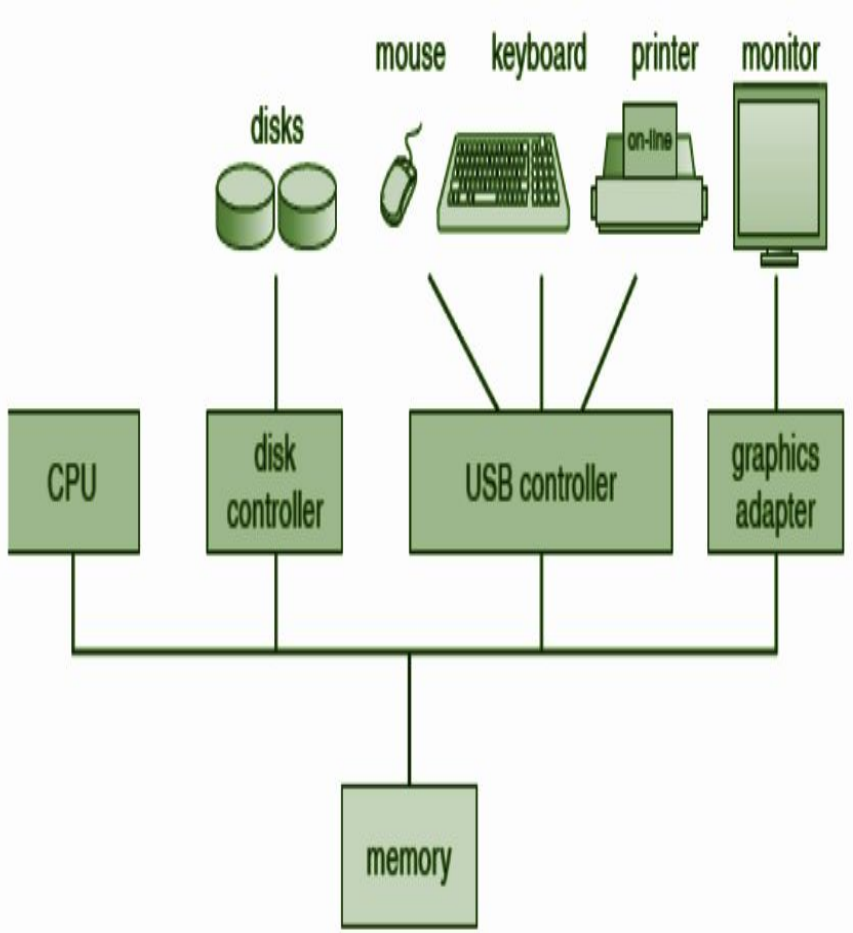
Computer System Operation

- When the system is **booted**, the first program that starts running is a **Bootstrap**.
- It is stored in read-only memory (**ROM**) or electrically erasable programmable read-only memory (**EEPROM**).
- **Bootstrap** is known by the general term **firmware**, within the computer hardware.
- It initializes all aspects of the system, from CPU registers to device controllers to memory contents.



Computer System Operation

- The bootstrap program must know how to load the operating system and how to start executing that system.
- The bootstrap program must locate and load into memory the operating system kernel.
- The first program that is created is **init**, after the OS is booted.
- It waits for the occurrence of event.



Common Functions of Interrupts

- **Interrupt (generated by the Hardware)** 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**

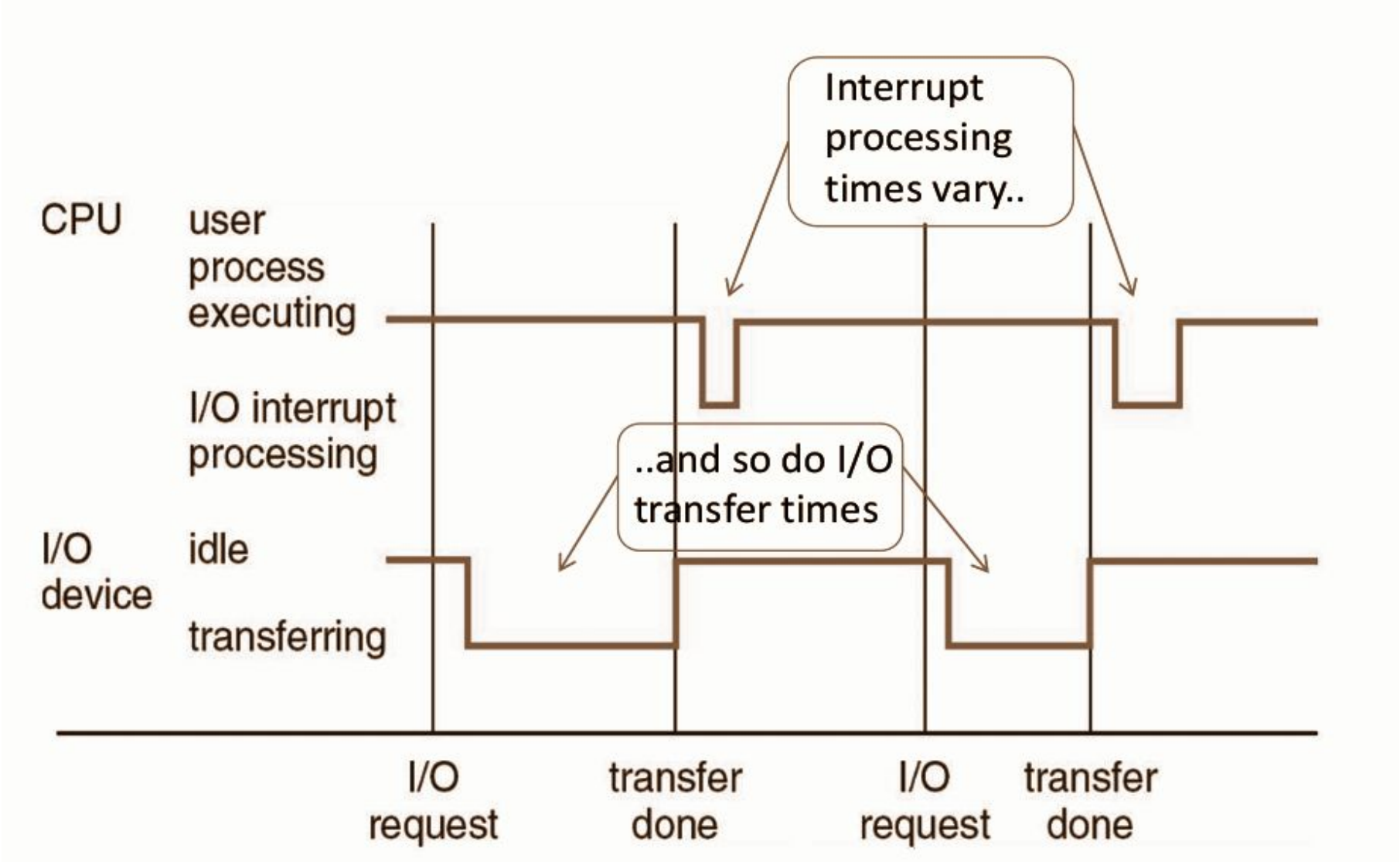
Interrupt Handling

- The operating system saves the state of the CPU by storing registers and the program counter

- Determines which type of interrupt has occurred: polling vectored interrupt system

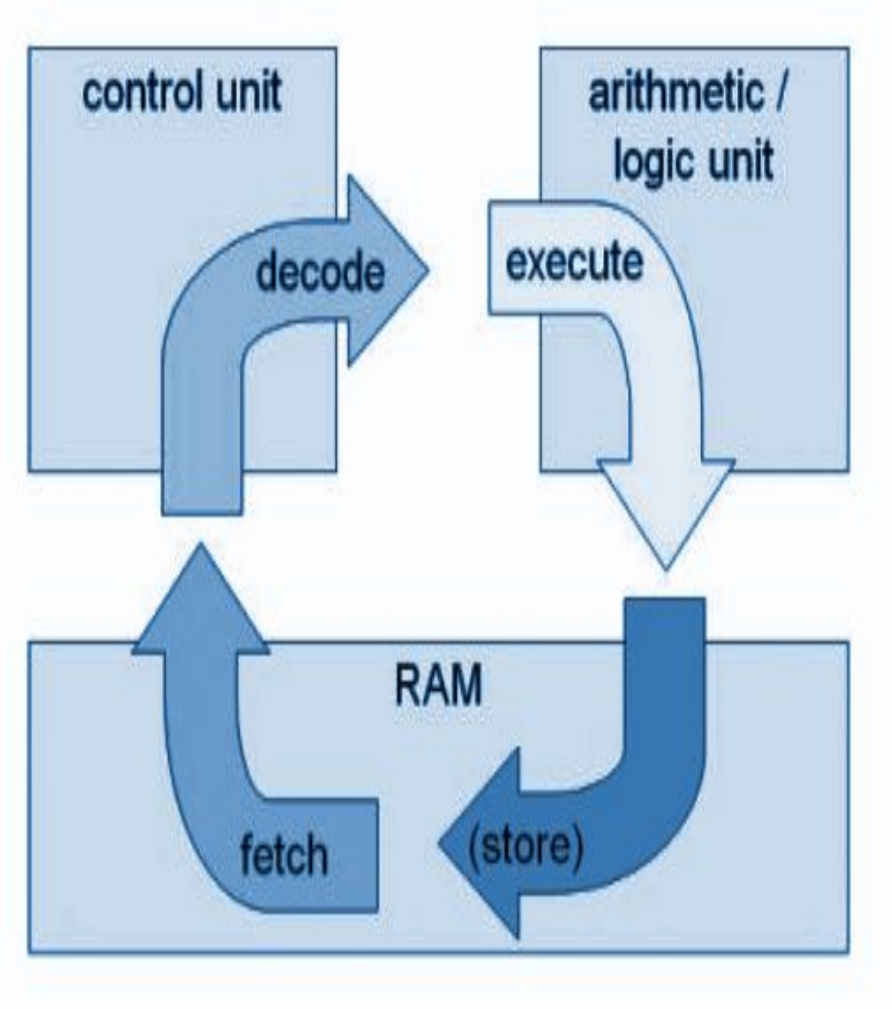
- Separate segments of code determine what action should be taken for each type of interrupt

Interrupt Timeline for a single process doing output



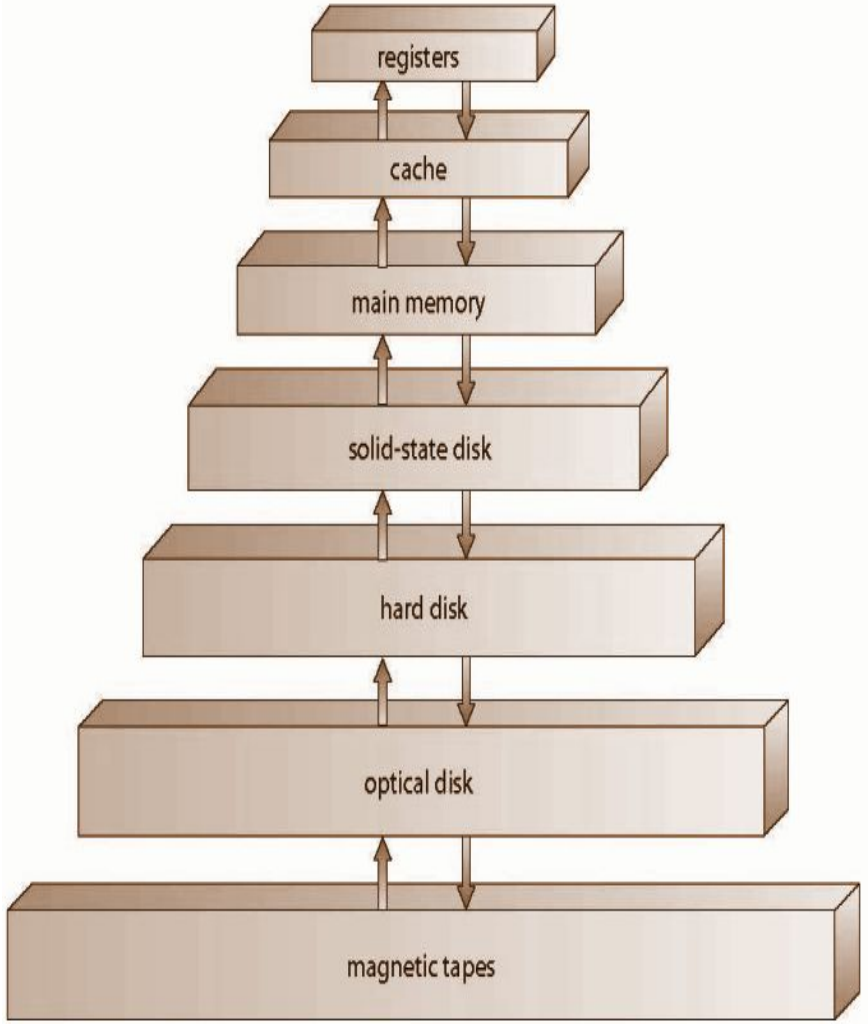
Program Execution Model

- The processor **fetches** instructions from memory, decodes and executes them.
- The **Fetch,Decode, and Execute** cycles are repeated until the program **terminates**.
- This is called the **Von Neumann model** of computing.



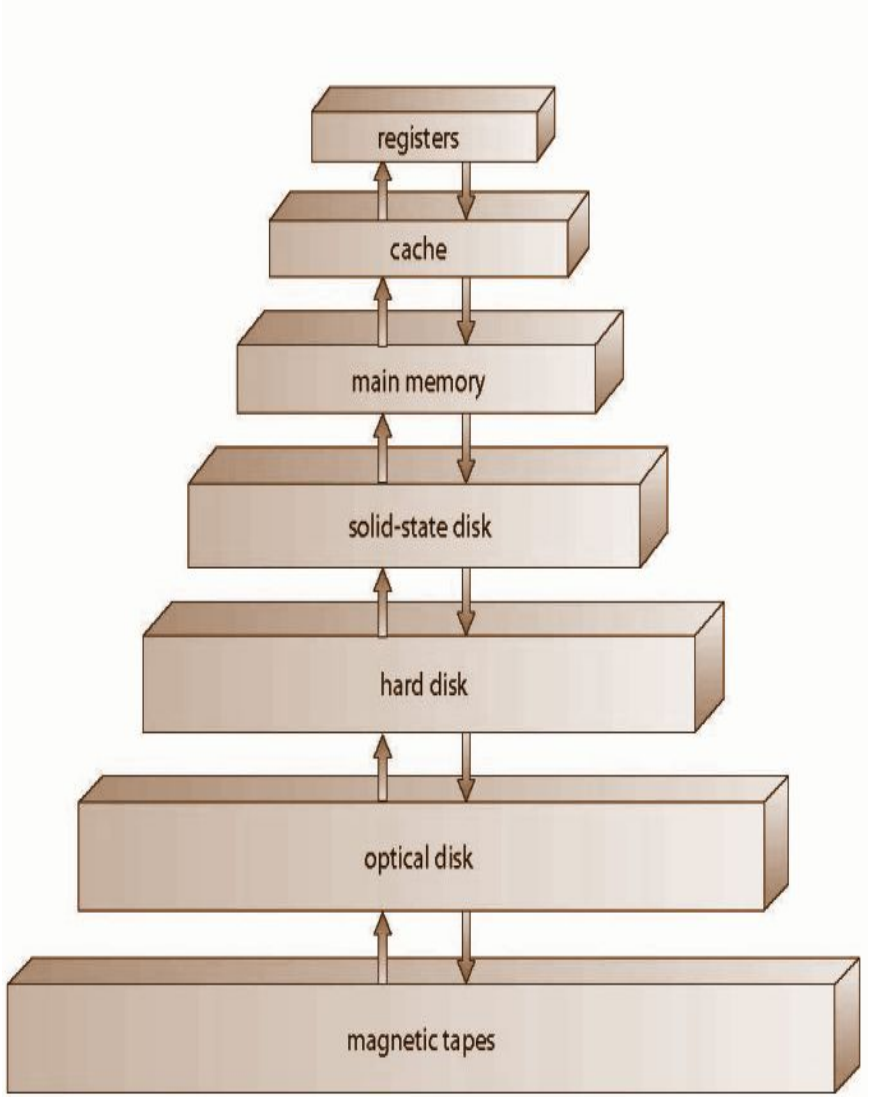
Storage Structure

- Main memory – only large storage media that the CPU can access directly
    - (Random access memory and typically volatile)
    - Implemented with semiconductor technology called DRAM
- Computers use other forms of memory like ROM, EEPROM
- Smart phones have EEPROM to store factory installed programs.



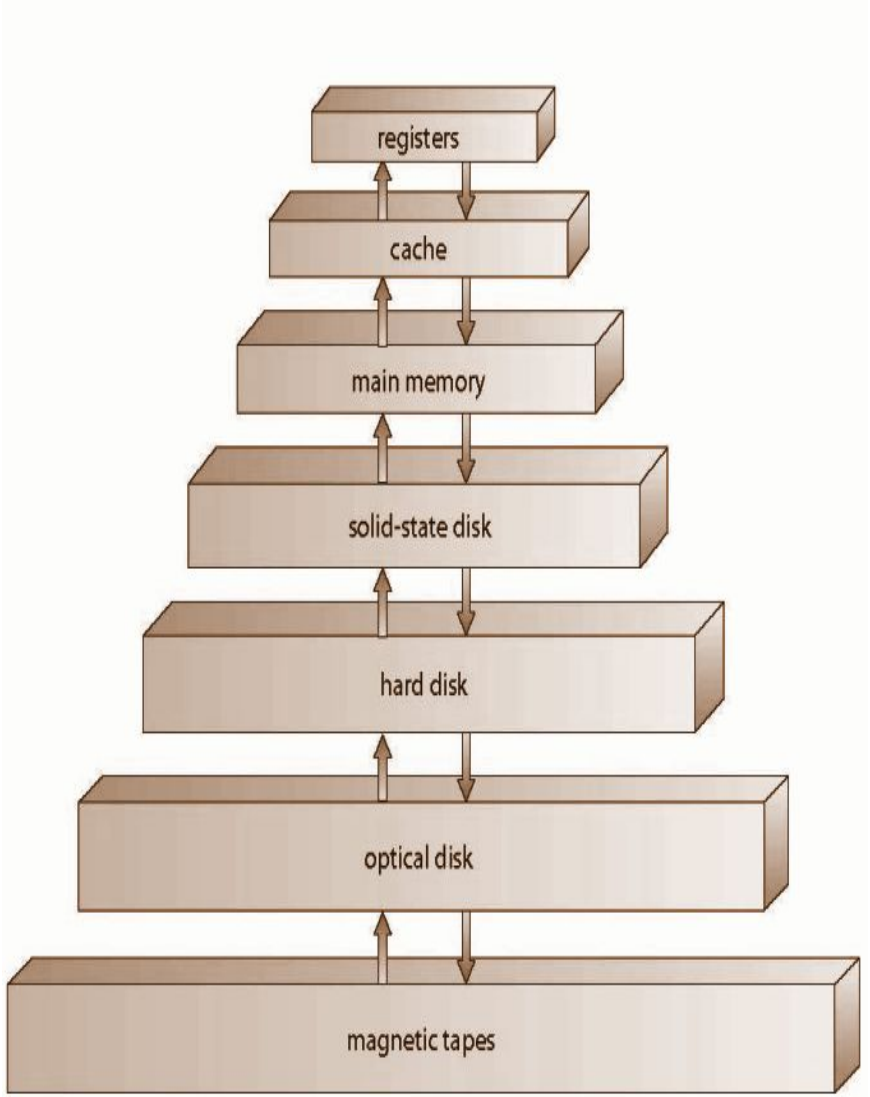
Storage Structure

- Secondary storage – extension of main memory that provides large nonvolatile storage capacity
- Hard disks – rigid metal or glass platters covered with magnetic recording material
- Solid-state disks – faster than hard disks, nonvolatile
- Storage systems organized in hierarchy
  - Speed
  - Cost
  - Volatility



Storage Structure

- 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, which provides uniform interface between controller and kernel



Caching

- Important principle, performed at many levels in a computer (in hardware, operating system, software)
- Information in use copied from slower to faster storage temporarily
- Faster storage (cache) checked first to determine if information is there
    - If it is, information used directly from the cache (fast)
    - If not, data copied to cache and used there
- Cache smaller than storage being cached
    - Cache management important design problem
    - Cache size and replacement policy

Typical Q and As

What is faster, Main Memory or Registers?	Registers are faster than main memory.
What is smaller, Main Memory or Registers?	Registers are smaller than main memory.
3 Types of registers typically available are:	Data Registers (storing data), Address Registers (stack pointer, frame pointer), Condition Codes (addl, mult)
The instruction execution consists of what two steps?	Fetch, Execute
Where does the processor fetch an instruction from?	Cache / Main Memory
Where does the processor hold the address of the next instruction?	Program Counter (PC)
Where does the processor hold the instruction fetched?	Instruction Register (IR)

Typical Q and As

When an external device becomes ready to be serviced by the processor, the device sends this type of signal to the processor:	Interrupt Signal
Two accepted methods of dealing with multiple interrupts is to:	Define priorities for the interrupts, Disable interrupts while an interrupt is being processed
A common class of interrupts is:	Program, Timer, I/O

OPERATING SYSTEMS

Typical Q and As

Hardware	CPU, Memory, I/O devices which provide basic computing resources for the system
Central Processing Unit (CPU)	Main Processor
Memory	Dynamic data storage unit
Input/Output devices	devices that are used to input to a computer (keyboard, mouse) or output (monitor)
Application Programs	Word processors, spreadsheets, compilers, and web browsers
Ease of Use	how convenient it is for a user to use a PC
Resource Utilization	how various hardware and software resources are shared
Mainframe	large and powerful data processing system

Typical Q and As

Systems program	associated with the OS but not part of the kernel
Bootstrap program	automatic procedure whereby basic OS is reloaded following a complete shutdown or loss of memory
Read-only memory (ROM)	read only memory
EEPROM	electrically erasable programable read only memory

Typical Q and As

Random Access Memory (RAM)	main memory, called random access memory
Von Neumann Architecture	Typical instruction-execution cycle where instructions are fetched from memory and stored in the instruction register
Instruction Register	Used to hold the current instruction that is being executed
secondary storage	extension of main memory where large quantities of data can be held permanently
Magnetic Disk	provides storage for both programs and data
Volatile storage	loses its contents when power to the device is removed
Nonvolatile storage	storage that does not lose its contents when power is removed
Electronic Disk	can be either volatile or nonvolatile, it stores data in large DRAM array

Topics Uncovered in this session

- Need for an Operating System
- Operating System Goals
- Why Study Operating Systems ?
- Computer System Structure
- What Operating Systems Do ?
- Computer System Organization

Topics Uncovered in this session

- Computer System Operation
- Common Functions of Interrupts
- Interrupt Handling
- Interrupt Timeline for a single process doing output
- Program Execution Model
- Storage Structure
- Typical Q and As

THANK YOU

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

nitin.pujari@pes.edu

For Course Deliverables by the Anchor Faculty click on www.pesuacademy.com