File Management

➢File Management

Q. **What is file?**

❖User view:

- File is a named collection of logically related information/data.
- File is a container which contains logically related information/data.
- File is a collection of characters/records/lines.
- File is a basic storage unit

❖**System view:**

- File is a **stream of bits/bytes.**

- **File = data + metadata**
  - ➢data = actual file contents
  - ➢metadata = information about the file.

- Data of the file is exists inside the file whereas information about the file gets stored inside one structure referred as **FCB (File Control Block).**
- In UNIX environment FCB is also called as an **iNode**.
- **FCB / iNode contains information about the file like:**
  1. iNode number: unique identifier of a file
  2. Name of the file
  3. Type of the file
  4. Size of the file
  5. Parent folder location
  6. Access permissions
  7. Time stamps etc...

- Per file one iNode/FCB gets created by the system and

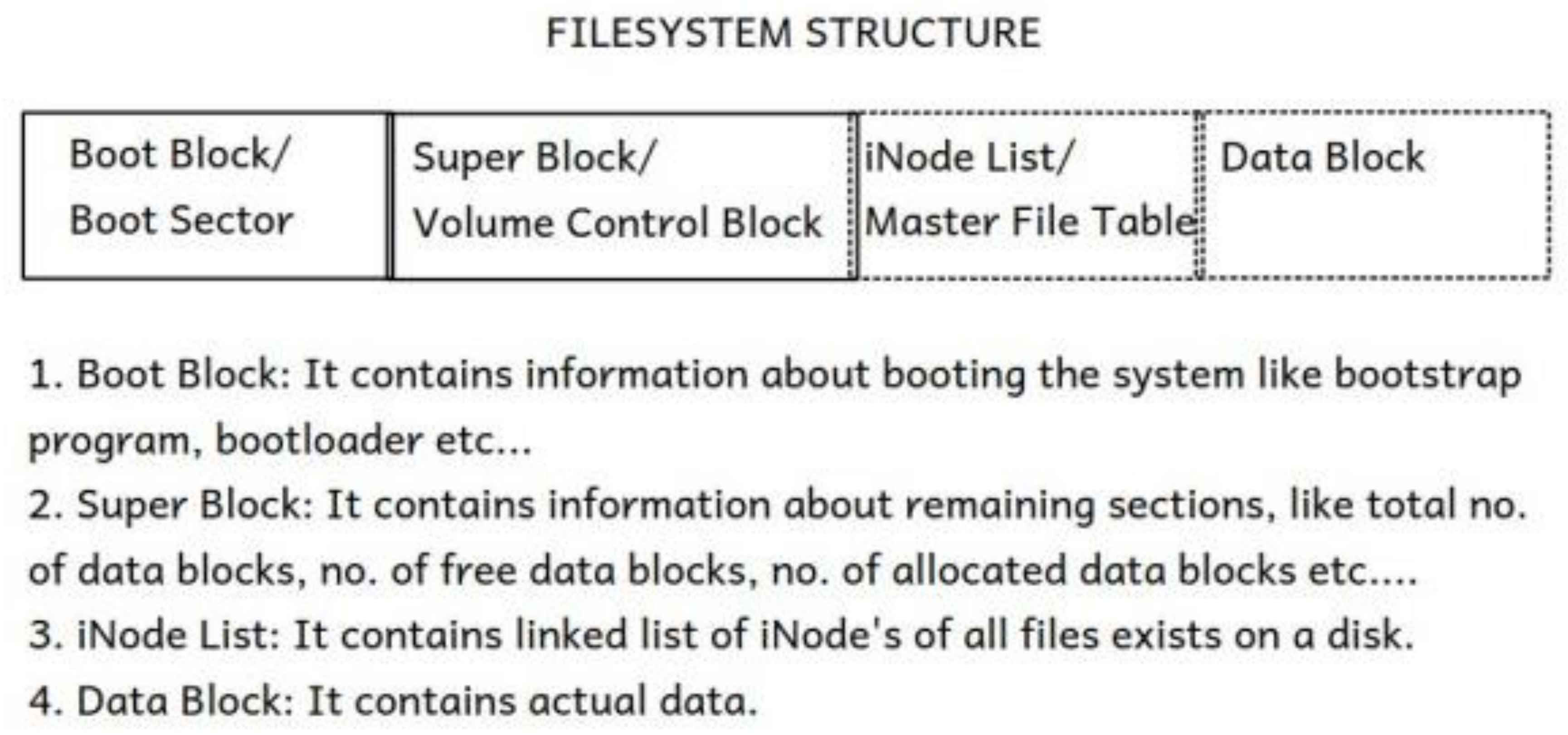
  -i.e. hence no. of iNodes = no. of files onto the disk.

- **data + metadata** of all the files are kept onto the disk, as disk may contents thousands of files, so thousands of iNodes and millions of bytes of data gets stored onto the disk, and hence all this **data + metadata** of all files need be keep onto the disk in an organized manner so that it can be accessed efficiently.

- **File system :** file system is a way to store data onto the disk in an organized manner so that it can be accessed efficiently and conveniently.
  - e.g. Each OS has its own file system like,
  - UNIX: UFS(UNIX File system),
  - Linux: Extended File system ext2, ext3, ext4,
  - Windows: FAT, NTFS etc...,
  - MAC OSX: HFS(Hierarchical File system) etc...

Operating Systems Concepts

- **Files system Structure:** File system divides disk/partition logically into sectors/blocks, like boot sector/boot block, volume control block/super block, master file table/iNode list block and Data Block.



FILESYSTEM STRUCTURE

Boot Block/ Boot Sector	Super Block/ Volume Control Block	iNode List/ Master File Table	Data Block

1. Boot Block: It contains information about booting the system like bootstrap program, bootloader etc...

2. Super Block: It contains information about remaining sections, like total no. of data blocks, no. of free data blocks, no. of allocated data blocks etc....

3. iNode List: It contains linked list of iNode's of all files exists on a disk.

4. Data Block: It contains actual data.

➢Disk space allocation methods:

- When a file is requesting for free data blocks, then in which manner free data blocks gets allocated for that file and how its information can be kept inside inode of that file is referred as disk space allocation method.

❖Three disk space allocation methods are there:

1. **Contiguous Allocation**
2. **Linked Allocation**
3. **Indexed Allocation**

1. **<u>Contiguous Allocation :</u>** free data blocks gets allocated for a file in a contiguous manner.

❑ **Advantages :**

1. Sequential access.
2. Random access .
3. Simple to implement.

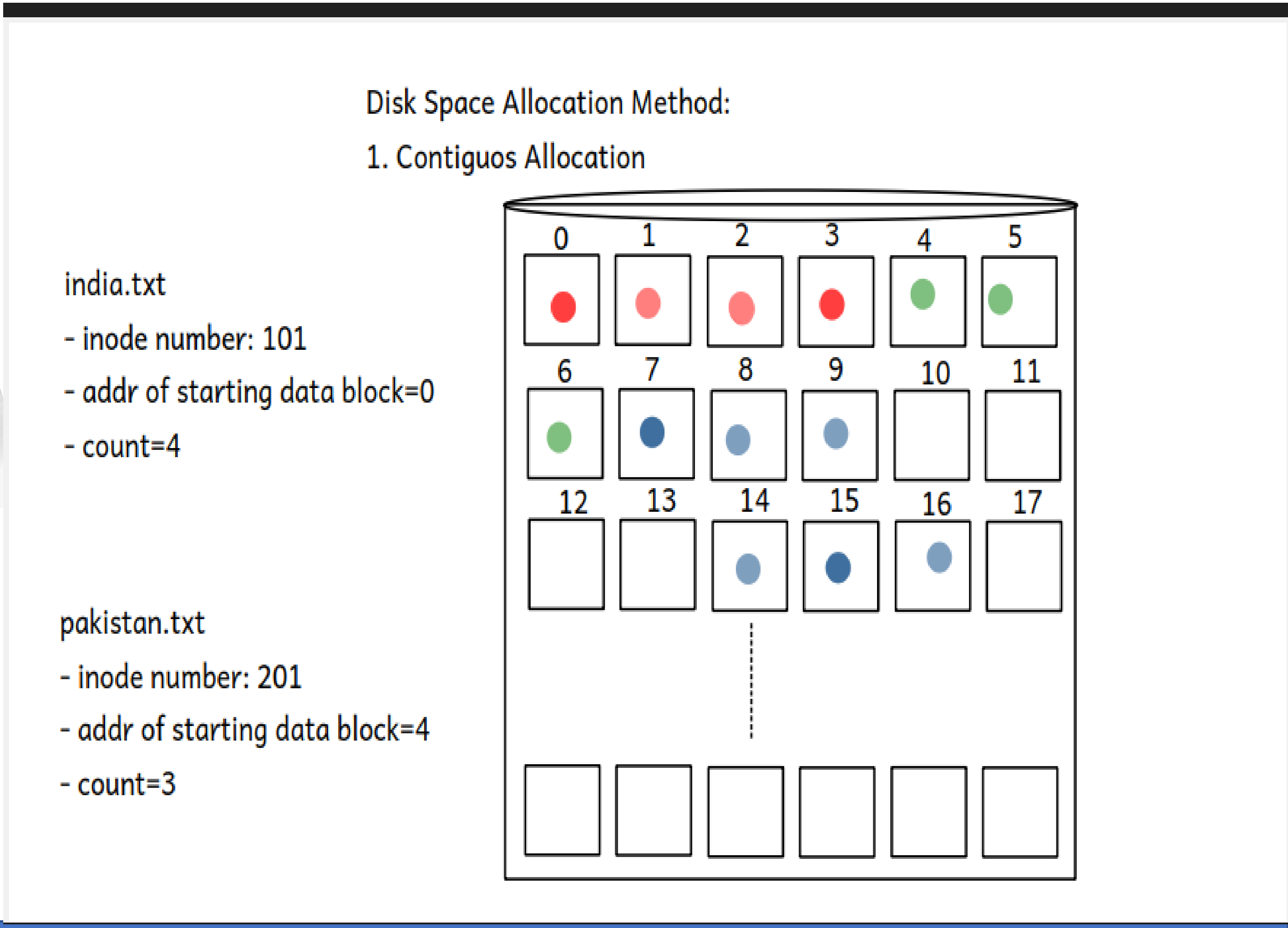
❑ **Disadvantages :**

1. File may not grow(Limitations).
2. **External fragmentation**

   -Number of blocks required are available but not contiguous.

❖ **Defragmentation :**

-Moves files on disk so that maximum contiguous free space is available.



Disk Space Allocation Method:

1. Contiguos Allocation

india.txt

- inode number: 101
- addr of starting data block=0
- count=4

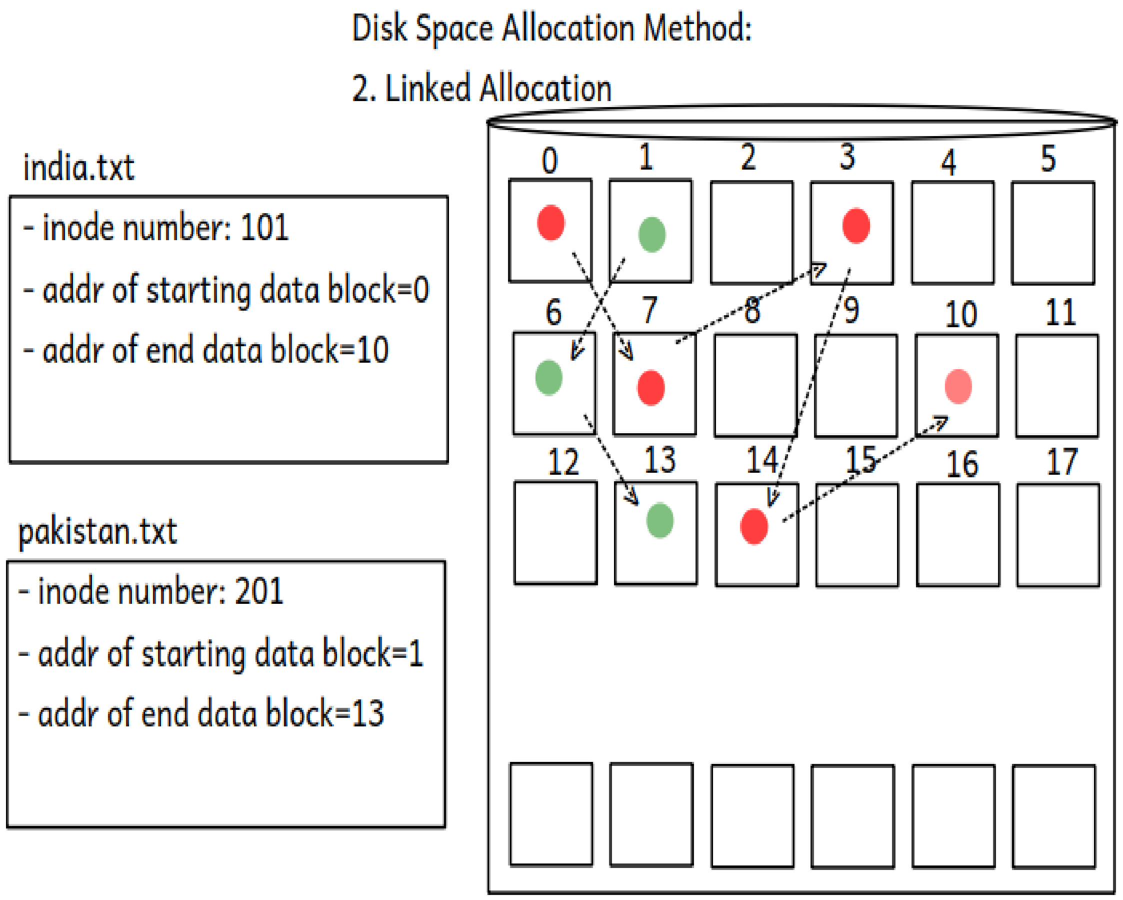
pakistan.txt

- inode number: 201
- addr of starting data block=4
- count=3

2. **<u>Linked Allocation:</u>** any free data blocks gets allocated for a file in a linked list manner.

- ❑ **Advantages :**
  1. Sequential access.
  2. No file grow limit.
  3. No external fragmentation.
- ❑ **Disadvantages :**
  1. Slow random access.

- ❑ **Example** : FAT



Disk Space Allocation Method:

2. Linked Allocation

india.txt

- inode number: 101
- addr of starting data block=0
- addr of end data block=10

pakistan.txt

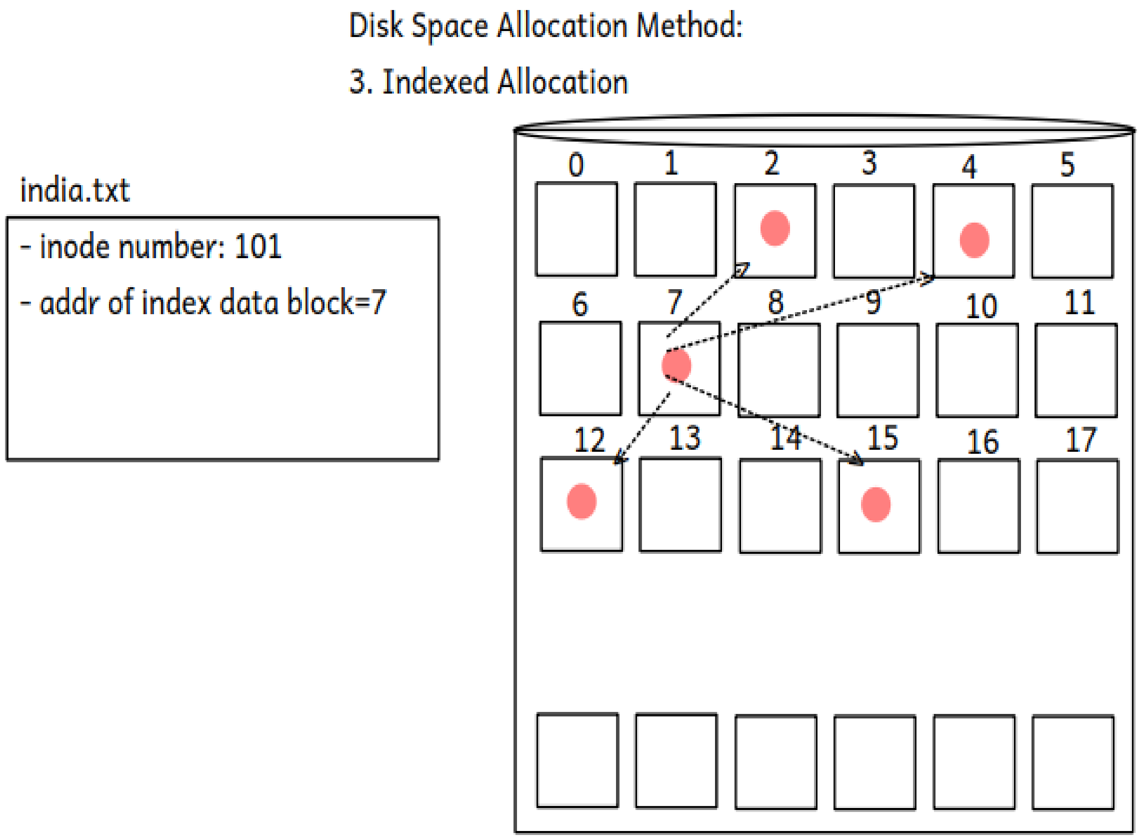
- inode number: 201
- addr of starting data block=1
- addr of end data block=13

3. **<u>Indexed Allocation :</u>** any free data blocks gets allocated for a file, as by maintaining an index data block information about allocated data blocks can be kept inside it.

- ❑ **Advantages :**
    1. Sequential Access.
    2. Random Access.
    3. No External Fragmentation

- ❑ **Disadvantages :**
    1. File cant grow up to some limit.

- ❑ **Example :** UFS , EXT 2 , EXT 3



Disk Space Allocation Method:

3. Indexed Allocation

india.txt

- inode number: 101

- addr of index data block=7

➢Disk Scheduling Algorithms:

- When system want to access data from a disk, request can sent to disk controller and disk controller accepts one request at a time and complete it.
- There are chances that at a time more than one requests for accessing data from the disk can be made by the processes running in a system, in that case all the requests can **be kept in a waiting queue of the disk maintained by an OS,** and there is need to schedule/select only one request at a time and sent it to the disk controller, to do this there are certain algorithms referred as **disk scheduling algorithms.**

1. **FCFS (First Come First Served):** request which is arrived first gets accepted and completed.
2. **SSTF (Shortest Seek Time First):** request which is closed to the current position of the head gets accepted and completed.
3. **SCAN:** head keeps scanning the disk from starting cylinder to end cylinder and whichever request came across gets accepted and completed.
4. **C-SCAN (Circular SCAN):** head scans the disk only in a one direction
5. **LOOK:** this policy can be used either with SCAN/C-SCAN, in this, if there is no request in a waiting queue then movement of the head gets stopped.

➢Memory Technologies

❖ There are four methods by which data can be accessed from the computer memory:

1. Sequential Access: e.g. Magnetic Tape
2. Direct Access: e.g. Magnetic Disk
3. Random Access: e.g. RAM Memory
4. Associative Access: e.g. Cache Memory

❖Magnetic Disk : Hard Disk Drive Structure

- HDD is made up of one or more circular platters arranged like CD rack.
- A Circular platter is a made up of non-magnetic substance like aluminum or aluminum alloy, which is coated with a magnetic substance.

➢Magnetic Disk : Hard Disk Drive Structure

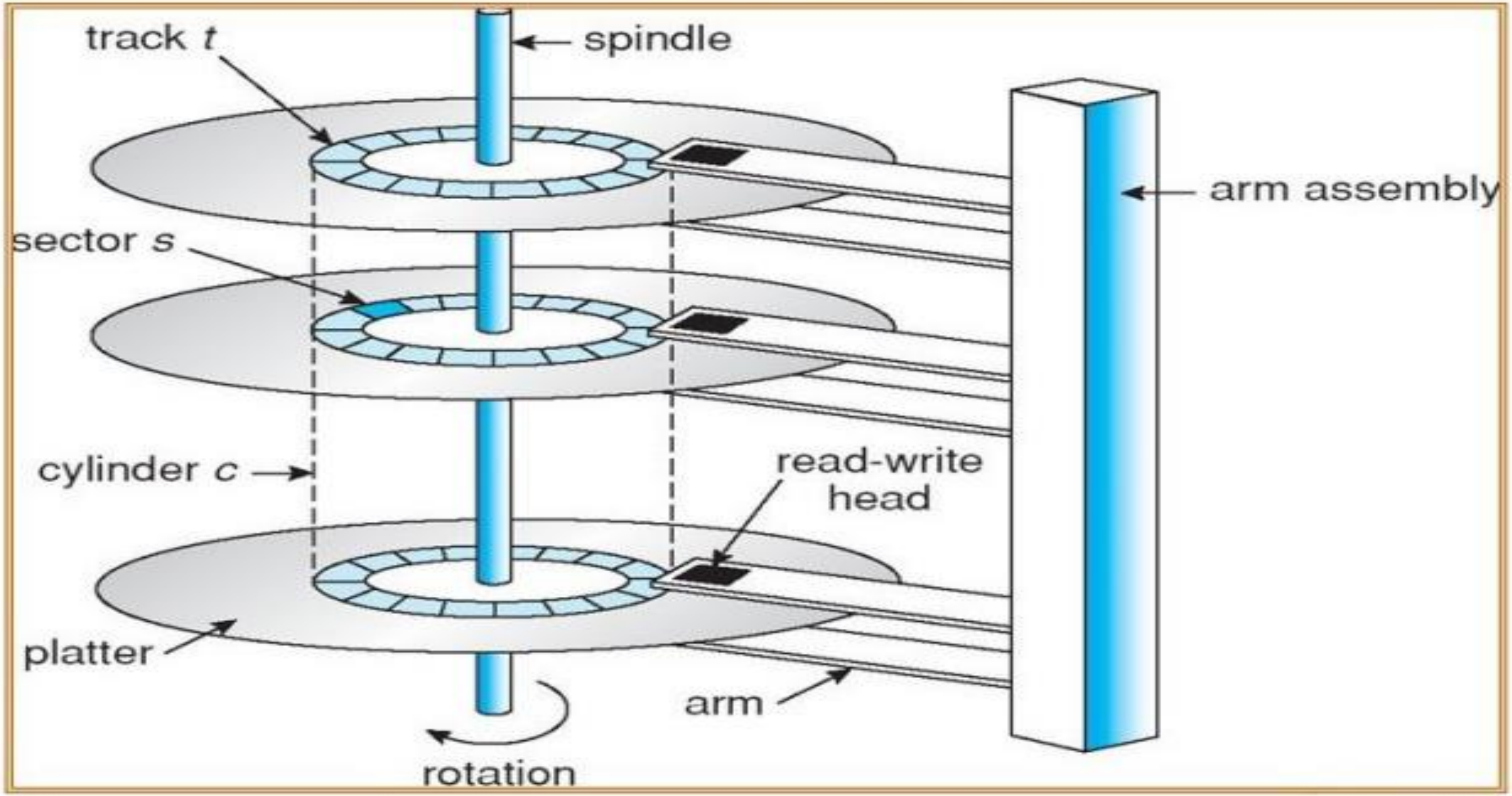
- **HDD** is made up **of one or more circular platters** arranged like CD rack.
- A Circular platter is a made up of **non-magnetic substance like aluminum or aluminum alloy**, which is coated with a magnetic substance.
- Coating of magnetic substance is either from one side to the platter or from both the sides (for increasing its capacity) and hence platter in a magnetic disk may be either single sided platter or double sided platter.
- **Circular platter is divided into the hundred's of concentric rings called as tracks** whereas **each track is divided into thousands of same size of blocks called as sectors.**
- Usually the **size of each sector is 512 bytes**
- There **is one conducting coil referred as head which is used to access data from the sector i.e. head can read and write data from and into a sector at a time.**
- **Head writes and read data sector by sector i.e. block by block, and magnetic disk is also called as block device.**

- All the operations like read, write, control etc... in a HDD are controlled by **disk controller**, and hence movement of the head also controlled by it.

- **Seek Time:** time required for the disk controller to move head from its current position to the desired track.
- **Rotational Latency:** after reaching head at desired track, circular platter gets rotated till the head does not comes aligned with the desired sector, and time required for this rotation is referred as rotational latency.

- **Access Time = Seek Time + Rotational Latency.**

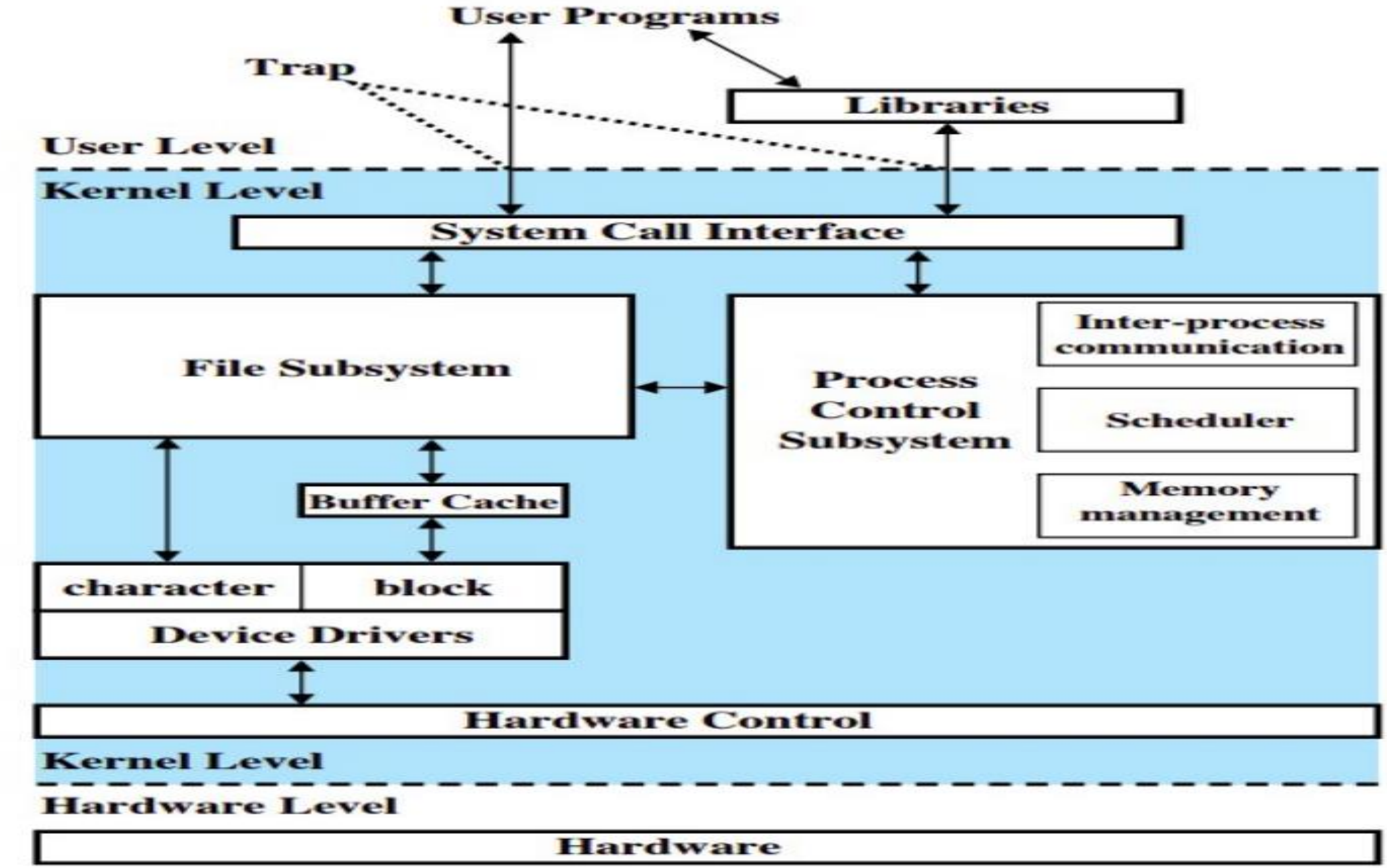
Moving-head Disk Machanism



UNIX Operating System:

- **UNIX** : UNICS – Uniplexed Information & Computing Services/System.
- **UNIX** was developed **at AT&T Bell Labs in US, in the decade of 1970’s by Ken Thompson, Denies Ritchie and team.**
- It was first run on a machine **DEC-PDP-7** (Digital Equipment Corporation Programmable Data Processing-7).
- UNIX **is the first multi-user, multi-programming & multi-tasking operating system**.
- UNIX was specially **designed for developers by developers**
- System architecture design of UNIX is followed by all modern OS’s like Windows, Linux, MAC OS X, Android etc..., and **hence UNIX is referred as mother of all modern operating systems.**

Operating Systems Concepts



Operating Systems Concepts

- Kernel acts as an interface between programs and hardware.
- Operating System has subsystems like System **Call Interface Block, File Subsystem Block, Process Control Subsystem Block (which contains IPC, Memory Management & CPU Scheduling), Device Driver, Hardware Control/Hardware Abstraction Layer.**
- **There are two major subsystems:**

  1. Process Control Subsystem
  2. File Subsystem

- In UNIX, **whatever is that can be stored is considered as a file** and **whatever is active is referred as a process.**
- **File has space & Process has life.**

- **System Calls:** are the functions defined in a C, C++ & Assembly languages, which provides interface of services made available by the kernel for the user (programmer user).

- If programmers want to use kernel services in their programs, it can be called directly through system calls or indirectly through set of libary functions provided by that programming language.
- There are 6 categories of system calls:

1. Process Control System Calls: e.g. fork(), _exit(), wait() etc...
   1. fork() : To create new processes
   2. _exit() : To exit processes
   3. wait() : To hold/wait processes
2. File Operations System Calls: e.g. open(), read(), write(), close() etc...
   1. open() :
   2. read() :
   3. write() :
   4. close() :

3. Device Control System Calls: e.g. open(), read(), write(), ioctl() etc...
4. Accounting Information System Calls: e.g. getpid(), getppid(), stat() etc...
    1. getpid() : To get Process ID
    2. getppid() : To get Parent Process ID
    3. stat() : To get File Information

5. Protection & Security System Calls: e.g. chmod(), chown() etc..
    1. chmod() : Change user mode / permission
    2. chown() : get file owner info

6. Inter Process Communication System Calls: e.g. pipe(), signal(), msgget() etc...

1. CUI/CLI : Command User Interface/Command Line Interface

- By using this kind of interface user can interacts with an OS by means entering commands onto the terminal/command line in a text format.
- e.g. In Windows name of the program which provide CUI => cmd.exe
- command prompt In Linux name of an application program which provides CUI => shell/terminal
- In MSDOS name of the program which provides CUI => command.com (Microsoft Disk Operating System).

2. GUI : Graphical User Interface

- by using this kind of interface user can interacts with an OS by means making an events like click on buttons, left click/right click/double click, menu bar, menu list etc.....
- - Windows = User friendly GUI.
- e.g. In Windows name of an application program which provides GUI => explorer.exe
- In Linux name of an application program which provides GUI => GNOME/KDE (GNU Network Object Model Environment / Common Desktop Environment).

Computer Fundamentals

➢Input Output Devices

- Devices which are connected to the motherboard externally through ports referred as peripheral devices or peripherals.
- An IO Devices are also referred as an external devices.

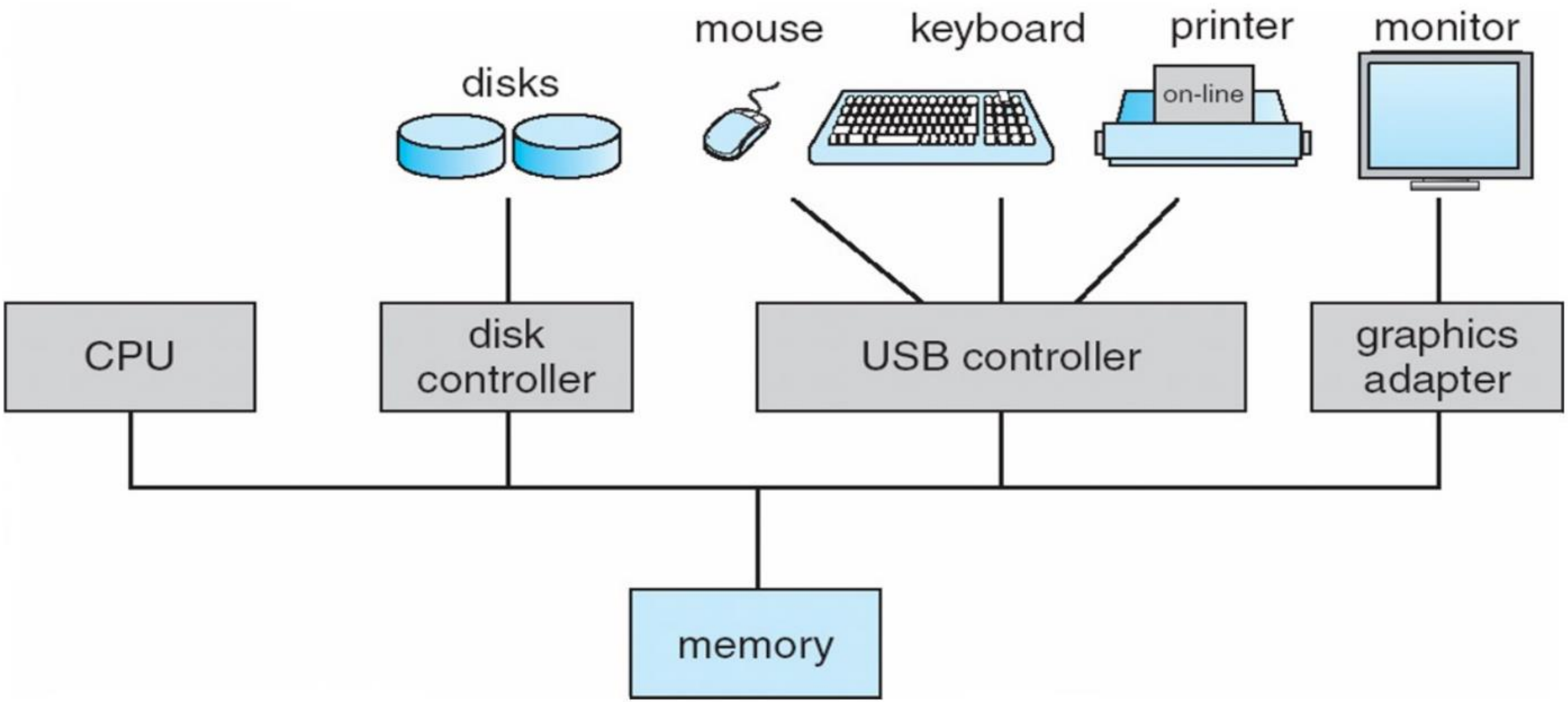
❖ Input Devices :

- Keyboard, Mouse, Scanner, Bar Code Reader, Eye Recognition System, Voice Recognition System, Touch Pad, Touch Screen etc...

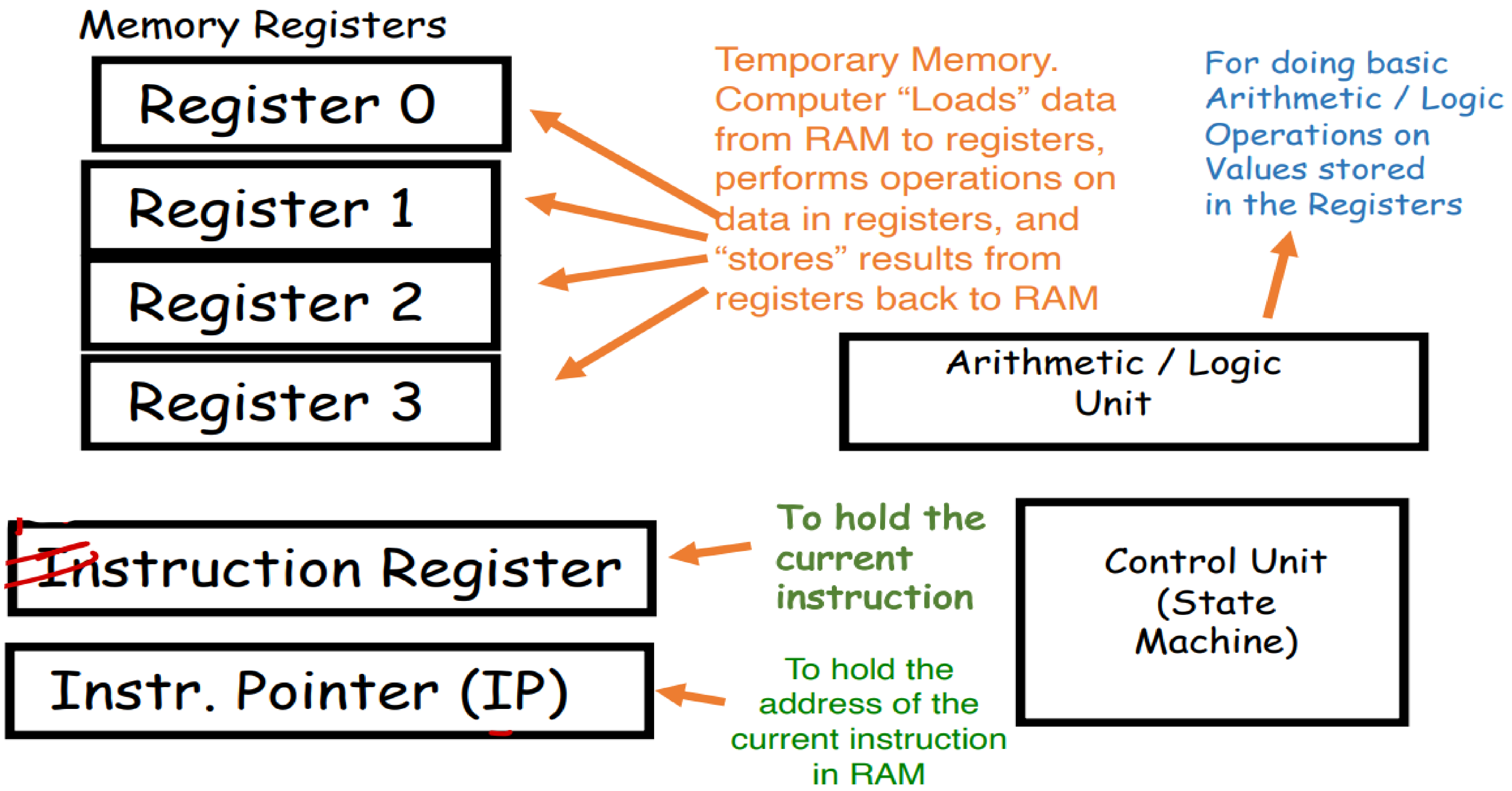
❖Output Device :

- Monitor, Printer, Speakers, Projector etc...

1. **Hardware** – provides basic computing resources (CPU, Memory, I/O devices, Communication).
2. **Operating System** – controls and coordinates use of the hardware among various application programs for various users.
3. **System & Application Programs** – ways in which the system resources are used to solve computing problems of the users (Word processors, Compilers, Web browsers, Database systems, Video games).
4. **Users** – (People, Machines, other computers).


mouse
keyboard
printer
monitor
disks
on-line
CPU
disk controller
USB controller
graphics adapter
memory

CPU



Memory Registers

Register 0

Register 1

Register 2

Register 3

Temporary Memory. Computer “Loads” data from RAM to registers, performs operations on data in registers, and “stores” results from registers back to RAM

For doing basic Arithmetic / Logic Operations on Values stored in the Registers

Arithmetic / Logic Unit

Instruction Register

To hold the current instruction

Instr. Pointer (IP)

To hold the address of the current instruction in RAM

Control Unit (State Machine)

Bus, CU, ALU, Memory

Bus

- It is a simplified way for many devices to communicate to each other.
- It is internal arrangement of computer system which includes design of the processor , memory and input/output units.

Control Unit

- Control is responsible for determining what action is to be performed on what data.
- controls all operations and it controls devices which are connected to the computer system by coordinating with device controllers.
- Fetch-Decode-Execute

ALU (Arithmetic Logic Unit)

- ALU is mainly comprised of **logic gates**, circuits made from transistors that take inputs.
- ALU performs all arithmetic and logical operations.

Memory

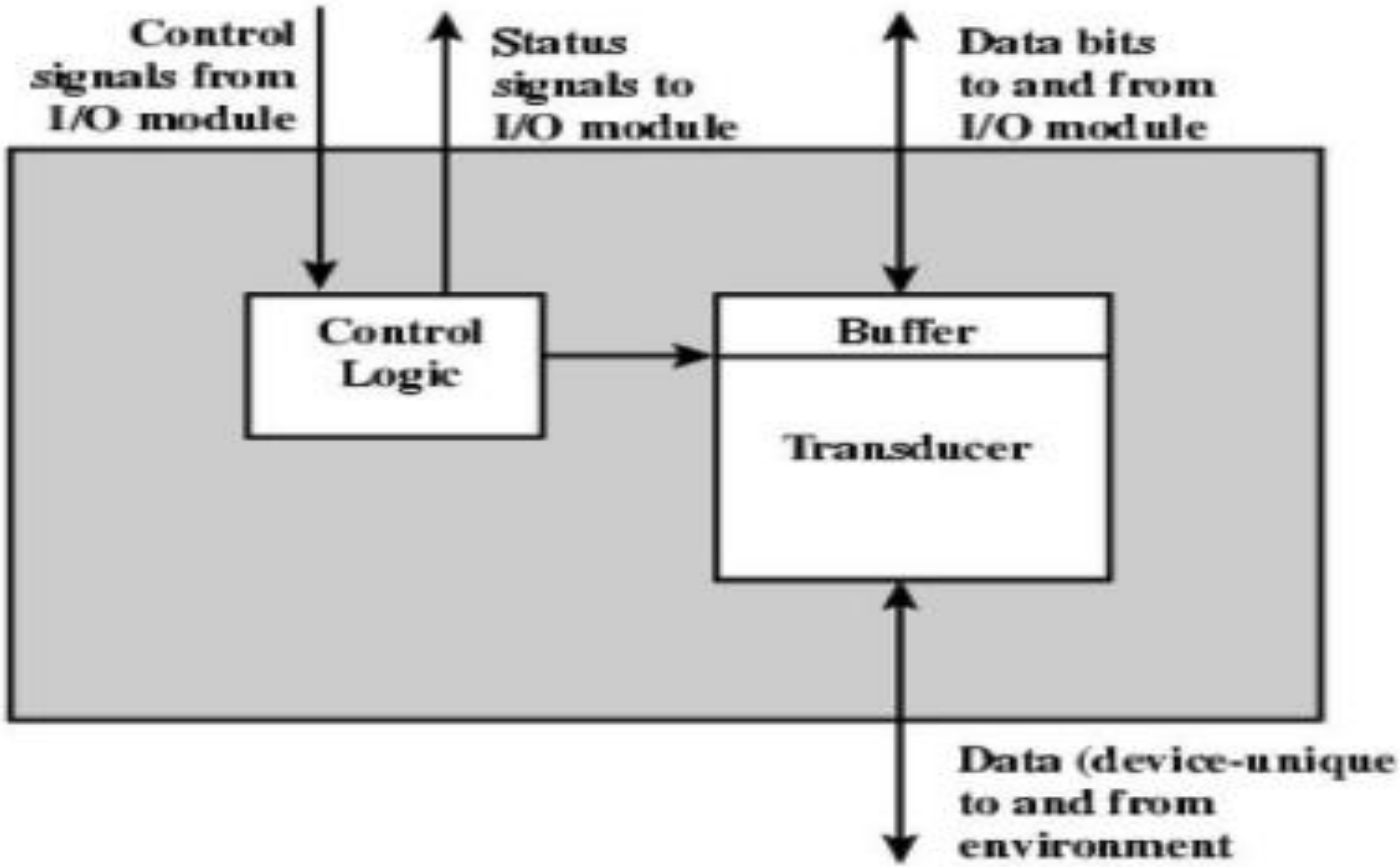
- Memory consists of circuits whose primary purpose is to **hold information**, but only temporarily.
- When you talk about the memory of a computer, most often you're talking about its RAM.

➢Structure of an External Device

- External Device has three major blocks:

1. **Control Logic Block(Controller):** controls all the operations of that device.
2. **Buffer:** each device has its own memory in which data can be stored temporarily referred as a buffer.
3. **Transducer:** this component converts any other form of energy into an electrical enery and converts an electrical energy into another form, this block of an external device is used to do communication with the outside world.

➢External Device Block Diagram :


Control signals from I/O module
Status signals to I/O module
Data bits to and from I/O module
Control Logic
Buffer
Transducer
Data (device-unique to and from environment

➢IO Modules/IO Ports:

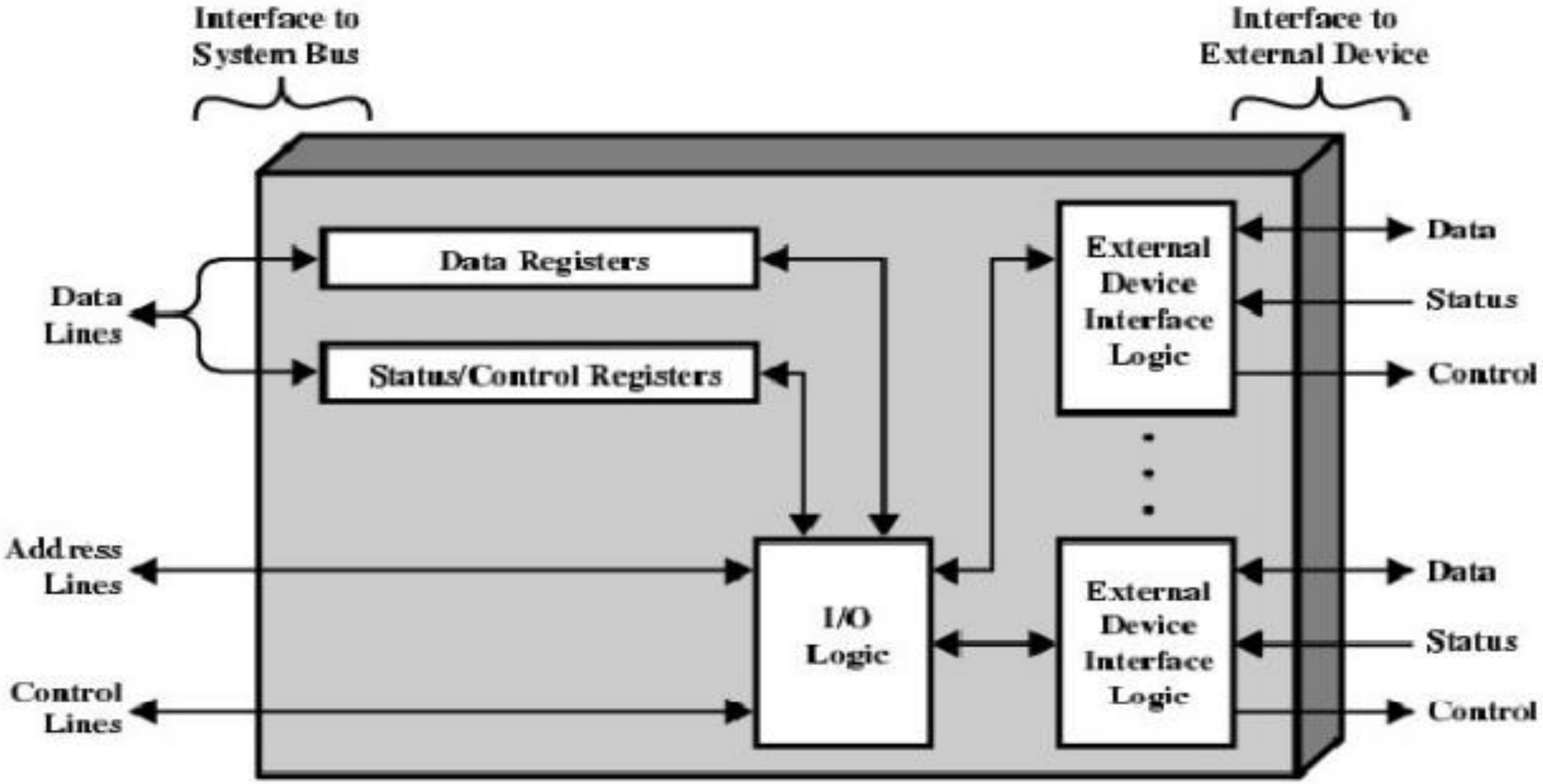
- Core Computer system is not able to communicates directly with any external device and hence I/O modules acts as an interface between core computer system and an I/O devices.
- Each IO device has its own internal dedicated processing unit called ' as **IO module**
- I/O Modules contains all the logic to communicates with an I/O devices.
- Single I/O module can be used for communication between one device or with more one devices as well

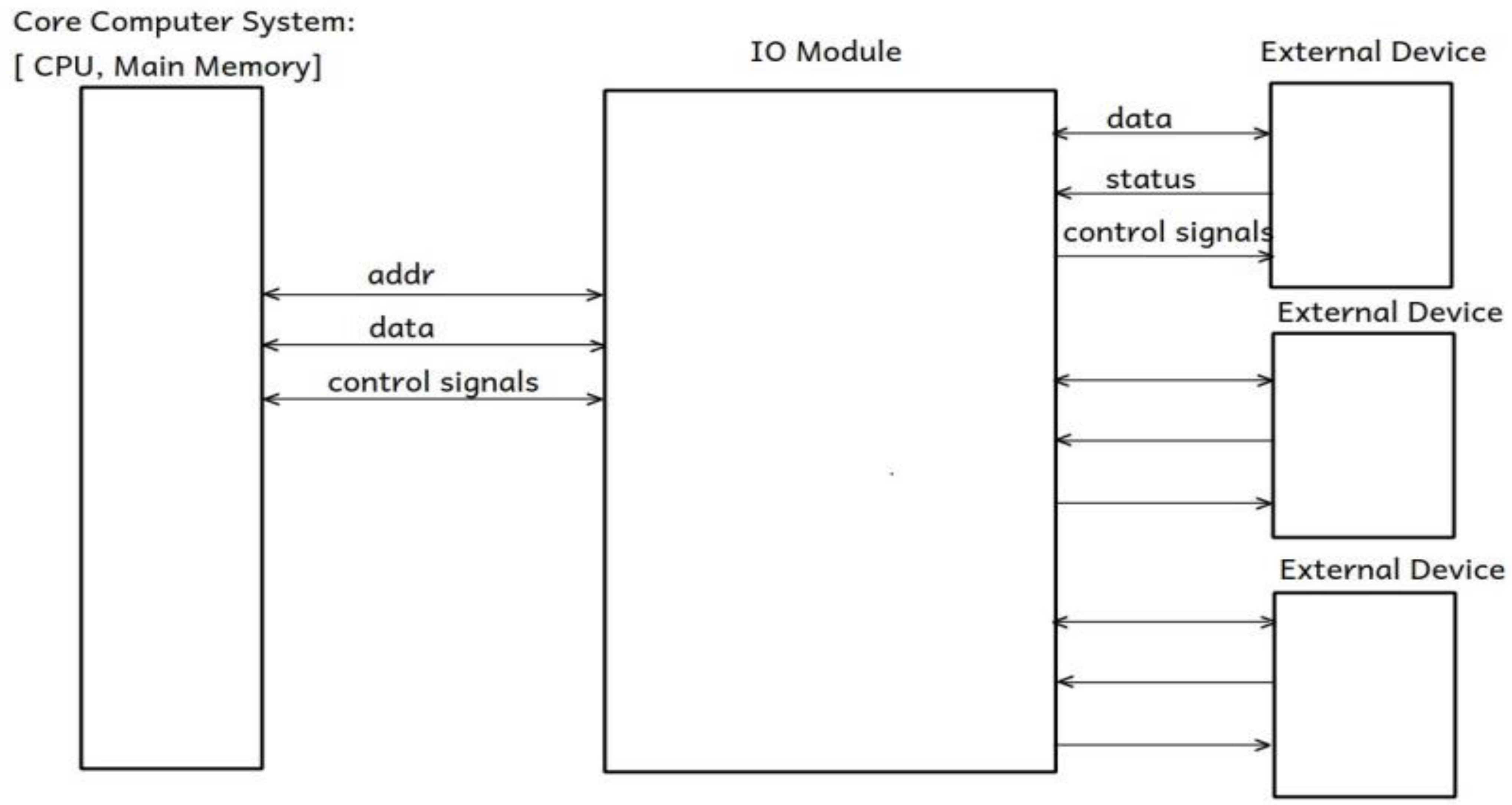
❖**Device Driver** - is a program within operating system that send/ receive data / command to/from I0 device controller and also handles interrupts send From the device

❖Functions Of IO Module :

1. **Control and Timing**
2. **CPU communication**
3. **Device communication**
4. **Data Buffering**
5. **Error Detection.**

➢I/O Module Structure Diagram :




Core Computer System:
[CPU, Main Memory]
IO Module
External Device
data
status
control signals
addr
data
control signals
External Device
External Device

➢Whenever there is transfer of data either from core computer system (i.e. Bus) to an IO devices or vice-versa, it is referred as an **I/O**.

❖**There are three IO techniques:**

1. Program driven IO

2. Interrupt IO

3. DMA i.e. Direct Memory Access

1. Program driven IO:

- All the logic/steps required for an I/O is there into one program, and by means of executing that program by the CPU I/O can be done.
- It done by Polling Method.

❖**Advantages:**

- Simple

❖**Disadvantages:**

- As the CPU remains wholly involved in an IO, less CPU utilization, and hence system performance is low.

2. Interrupt IO:

❖What is an interrupt?

- An interrupt is a signal received by the CPU due to which it stops an execution of one job and starts an execution of another job.

❖Advantages:

- In this IO, the CPU remains involved in an IO whenever gets interrupted, and hence its utilization can be maximized.

❖Disadvantages:

- When there is a data transfer between main memory & secondary memory unnecessary involvement of the CPU is there.

❖Interrupt Handling :

- Interrupt handling function is also called as **interrupt service rutine(ISR).**
- There are many IO devices in system , so there interrupt handling process also different like keyboard , mouse , disk etc..
- So there are Table maintain by OS that store all device interrupt function starting address . This table is called **as Interrupt Vector Table(IVT).**

3. DMA (Direct Memory Access):

- Whenever there is a transfer of data between core computer system and IO devices (e.g. main memory and secondary memory), the CPU initiates an IO and gives control of an IO process to the DMA controller, and hence onwards that IO process is controlled by the DMC controller till the end i.e. the DMA controller will work on behalf of the CPU and after finishing an IO it sends acknowledgement to the CPU, and by the time the CPU can execute another jobs, and utilization of the CPU can be maximized further.
- e.g. 8237 DMA controller

Thank you!

Kiran Jaybhave

email – kiran.jaybhave@sunbeaminfo.com