Operating System Concepts

SunBeam Institute of Information & Technology, Hinjwadi, Pune & Karad.

Introduction

Mr. Sachin Gajanan Pawar.
Lead Technical @SunBeam.

Educational Qualification:

1. B.Sc. Electronics.
2. M.Sc.Tech (Industrial Maths & Computer Applications).

Work Experience:

1. 7 Yrs of teaching experience in SunBeam Pune & Karad, for PreCAT & CDAC Courses.
2. 3+ Yrs of Industry Experience of Software Development.

Skills:

1. C Programming Language
2. C++ Programming Language
3. Data Structures
4. Operating System Concepts

*** Introduction:**

- Why there is need of an OS?
- What is an OS?
- Functions of an OS

*** Computer Fundamentals:**

- Major Components: Processor, Memory Devices & IO Devices.
- Memory Technologies and its characteristics
- IO Techniques

*** UNIX System Architecture Design**

- Major subsystem of an UNIX system: File subsystem & Process Control subsystem.
- System Calls & its catagories
- Dual Mode Operation

*** Process Management**

- What is Process & PCB?
- States of the process
- CPU scheduling & CPU scheduling algorithms
- Inter Process Communication: Shared Memory Model & Message Passing Model

* **Process Management**
  - Process Synchronization/Co-ordination
  - Deadlocks & deadlock handling methods

* **Memory Management**
  - Swapping
  - Memory Allocation Methods
  - Internal Fragmentation & External Fragmentation
  - Segmentation
  - Paging
  - Virtual Memory Management

*** File Management**

- What is file?
- What is filesystem & filesystem structure?
- Disk space allocation methods
- Disk scheduling algorithms

Q. Why there is a need of an OS?

- Computer is a machine/hardware does different tasks efficiently & accurately.

- Basic functions of computer:

  1. Data Storage
  2. Data Processing
  3. Data Movement
  4. Control

- As any user cannot communicates/interacts directly with computer hardware to do different tasks, and hence there is need of some interface between user and hardware.

Operating System Concepts

User(Human beings, Machine, Other Computer)

Programs (User/Application/System)

e.g. notepad, wordpad, browser, device driver etc....

Operating System

System Software(i.e. collection of system programs)

Hardware(CPU, Memory, IO device)

Q. What is an Operating System?

- An OS is a **system software** (i.e. collection of system programs) which acts as an interface between user and hardware.

- An OS also acts as an interface between programs and hardware.

- An OS allocates resources like main memory, CPU time, i/o devices access etc... to all running programs, hence it is also called as a **resource allocator.**

- An OS controls an execution of all programs and it also controls hardware devices which are connected to the computer system and hence it is also called as a **control program.**

Q. What is an Operating System?

- An OS manages limited available resources among all running programs, hence it is also called as a **resource manager.**

- From End User: An OS is a software (i.e. collection of programs) comes either in CD/DVD, has following main components:

1. Kernel: It is a core program/part of an OS which runs continuosly into the main memory does basic minimal functionalities of it.

e.g. Linux: vmlinuz, Windows: ntoskrnl.exe

2. Utility Softwares: e.g. disk manager, windows firewall, anti-virus software etc...

3. Application Softwares: e.g. google chrome, shell, notepad, msoffice etc...

Functions of an OS:

Basic minimal functionalities/Kernel functionalities:

1. Process Management
2. Memory Management
3. Hardware Abstraction
4. CPU Scheduling
5. File & IO Management

Extra utility functionalities/optional:

6. Protection & Security
7. User Interfacing
8. Networking

Computer Fundamentals:

- Computer is a hardware mainly contains:

1. Processor

2. Memory Devices

3. IO Devices

- There are two basic/fundamental **structural** and **functional units** of computer hardware system:

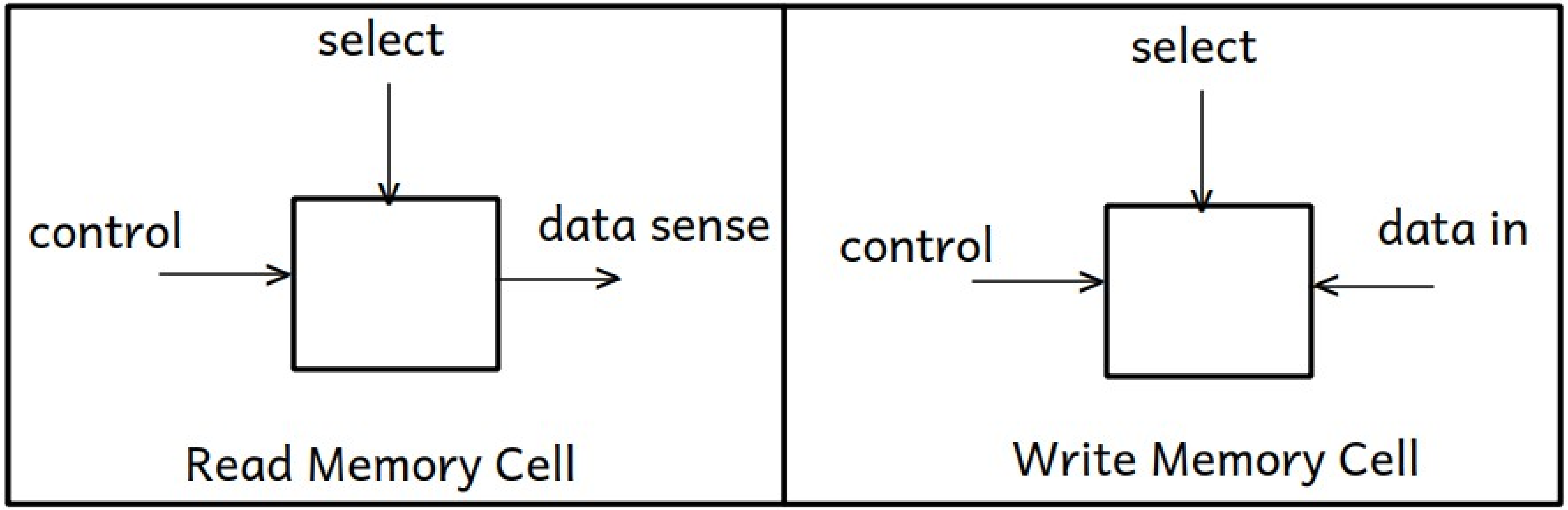
1. Memory cell

2. Gates

- As the data stoarage can be acheived by memory cell, data processing, data movement and control functions can be achieved by using gates, hence memory cell & gates are referred as functional units, wheras computer is a digital device made up of collection of millions of memory cells & gates, and hence memory cell & gates are referred as structural units as well.

Computer Fundamentals:

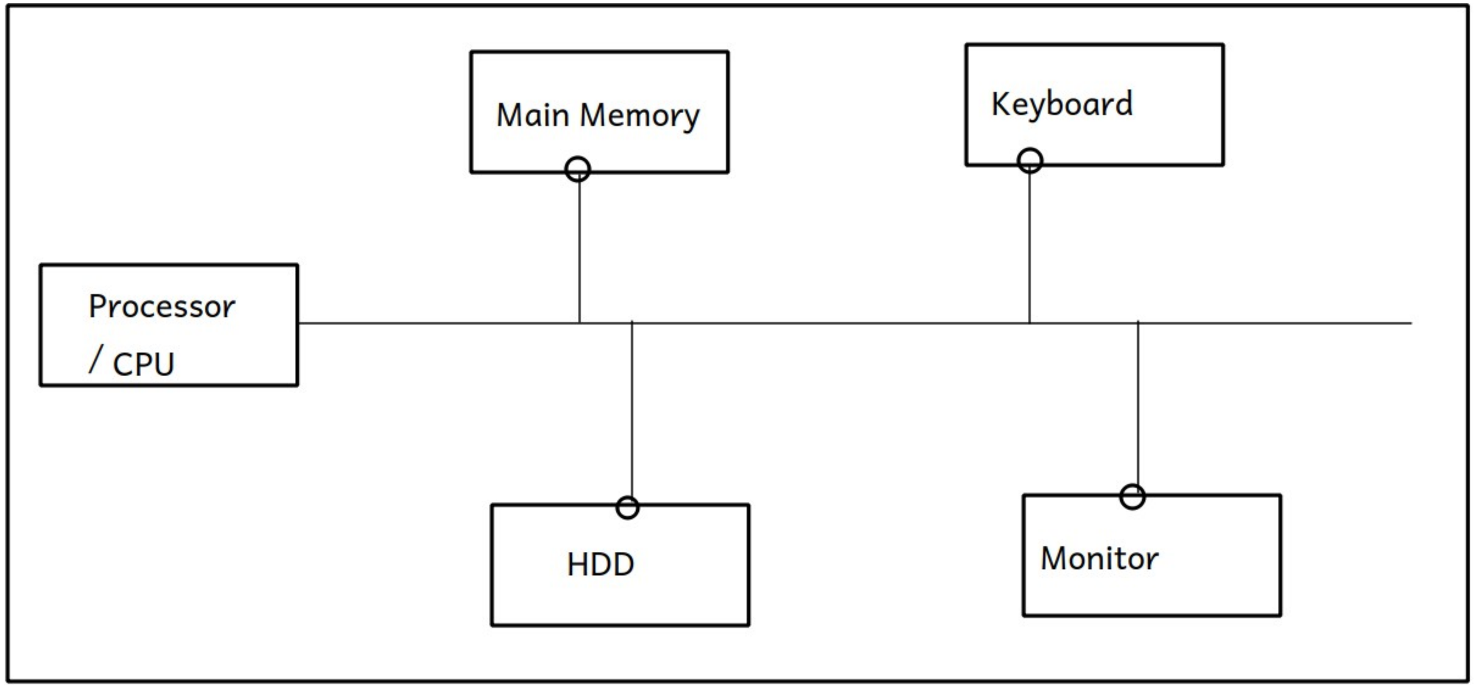
1. Memory cell: memory has the capacity to store one value at a time in it either 0 OR 1.



2. Gates: gate a basic digital device (flip-flop gates) can be used for performing arithmetic & logic operations as well as for control.

Processor:

- Processor contains: ALU(Arithmetic Logic Unit), CU(Control Unit), Registers.
- Processor is also called as **CPU(Central Processing Unit).**



Operating System Concepts

- Computer system mainly contains Processor, Main Memory, Hard Disk Drive, Keyboard & Monitor.

- In a Computer system each and every device has got its own dedicated processor which controls operations of that device specifically, and there exists one processor which controls all the operations and devices in a computer system centrally by co-ordinating with dedicated processors.

For Example: Hard Disk Drive has **disk controller** which controls all disk operations, whereas processor in a block diagram controls all operations

- All components in a computer are connected via conducting wires through which transfer of data, addresses and control signals takes place, these conducting wires are referred as **buses.**

- Three types of buses are there:

1. Data Bus: transfers data

2. Address Bus: transferes addresses

3. Control Bus: transferes control signals

- Major components of computer system (i.e. components which are onto the motherboard like CPU, Cache Memory, Main Memory etc...) are connected via a bus referred as a **system bus**.

- As data which is sent by one component can be recieved by any other component connected to the bus, hence it is a **shared communication pathway.**

- Control signals sent by the CPU to other devices referred as commands.

1. TEST: to check the status of any device

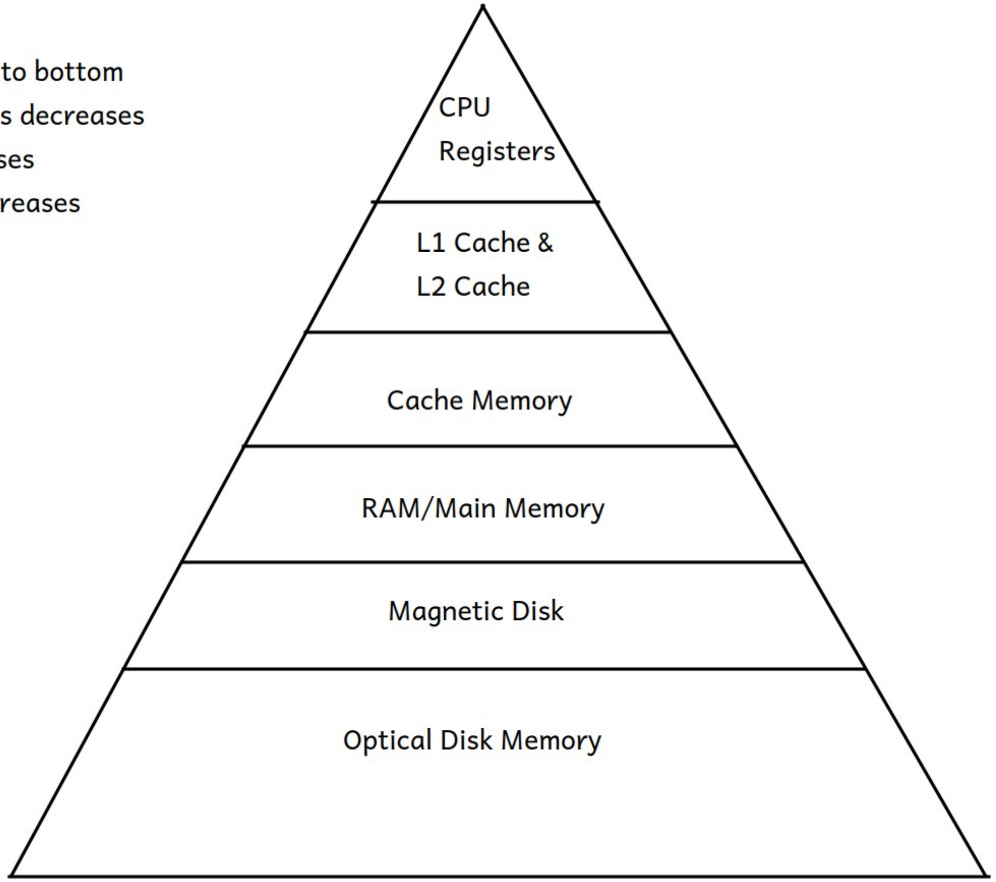
2. WRITE

3. READ

4. CONTROL

Computer Memory Technologies:

as we go from top to bottom
- access speed gets decreases
- cost gets decreases
- capacity gets increases



Computer Memory Hierarchy

Computer Memory Technologies:

- **CPU Registers:** memory which is very close to the CPU are registers which is at the top in a computer memory heirarchy.

- Instructions and data currently executing by the CPU can be kept temporarily into the CPU registers.

- MAR, MBR, IOAR, IOBR, PC, SP, Accumulator etc...

- Computer memory can be catagorised into two catagories as per its location: **Internal Memory & External Memory.**

- Internal Memory: memory which is internal to the motherboard is referred as an internal memory.

e.g. CPU registers, L1 & L2 cache Cache memory, RAM.

- External Memory: memory which is external to the motherboard is referred as an external memory.

e.g. magnetic disk, optical disk, magetic tape etc...

 # Computer Memory Technologies:

- Computer memory can also catagorised into two catagories: **Primary Memory & Secondary Memory.**

- Primary Memory: memory which can be accessible directly by the CPU is referred as primary memory, i.e. memory which can accessible by the CPU with the help of instruction set having with the CPU.

e.g. CPU registers, L1 & L2 Cache, Cache Memory, RAM

- Secondary Memory: memory which cannot be accessed directly by the CPU is referred as secondary memory.

e.g. Magetic Disk, CD/DVD, PD etc..

- If the CPU want to access disk contents, first it gets fetched into the RAM and then it can be accessed by the CPU from RAM.

- As for an execution of every program RAM memory is must and hence RAM is also called as **Main memory**.

Q. Why there is a need of cache memory?

- As the rate at which the CPU can execute instructions is faster than the rate at which data can be accessed from the main memory, so even the CPU is very fast, with the same speed data do not gets fetched from the main memory for execution, hence due to this speed mismatch overall system performance gets down.

- To reduce speed match between the CPU and the main memory Cache memory (hardware) can be added between them and system performance can be increases by means reducing speed mismatch.

Q. What is Cache Memory?

- Cache memory is **faster** memory, which is a type RAM i.e. **SRAM**, in which **most recently accessed main memory contents** can be kept/stored in an **associative manner i.e. in a key-value pairs.**

- There are two types of RAM:

1. DRAM (Dynamic RAM): memory cells are made up of capacitors

- Main memory is as example of DRAM.

2. SRAM (Static RAM): memory cells are made up of flip-flop gates

- Cache Memory is an example of SRAM.

- Cache Memory has **C** no. of lines, whereas each line is divided into two parts, each line contains k words of data (rcently accessed main memory contents) and its main memory addresses can be kept in few tag bits.

1. First part of a line : **few tag bits** contains main memory addresses of k words of data in that line

2. Second part of a line contains k words of data.

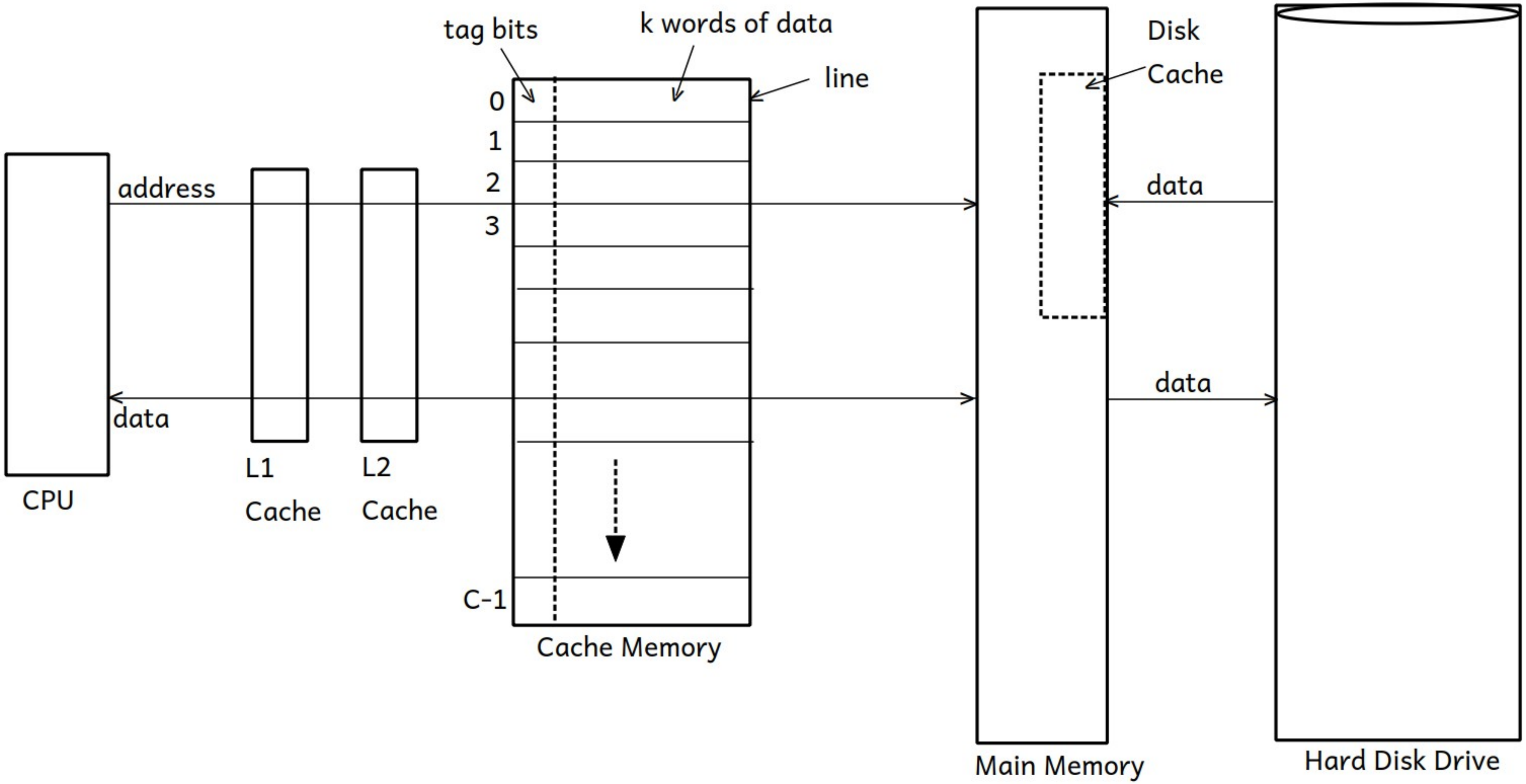
- When the CPU want to fetch data from the main memory it requests for its address, and this requested address gets searched into the cache memory first, if requested addr is found in the cache memory then data also found in a cache memory, it is referred as **cache hit**, whereas if the requested address and hence data is not found in a cache memory then it is referred as a **cache miss,** in that data gets fetched from main memory and gets transferred to the CPU via cache memory only.

- Even after adding cache memory between the CPU and main memory, the rate at which the CPU can execute instructions is faster than the rate at which at which data can be accessed from cache memory, and hence to reduce speed mismatch between the CPU and cache memory one or more levels of cache memory i.e. **L1 cache & L2 cache** can be added between them.

- **Disk Cache:** it is **purely a software technique** in which portion of the main memory can be used as a cache memory in which most recently accessed disk contents can be kept in an associative manner, so whenever the CPU want to access data from hard disk drive it first gets searched into the disk cache.

- Disk cache technique is used to reduce speed mismatch between the CPU and Secondary memory.

Operating System Concepts



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

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

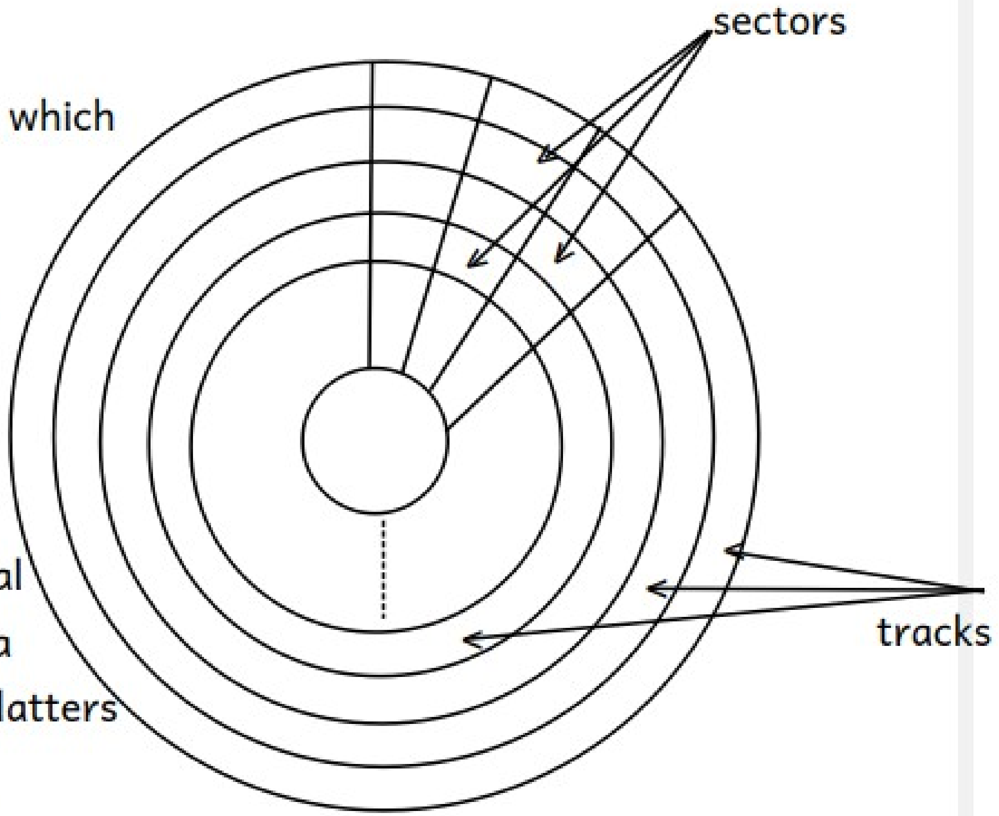
Magnetic Disk : Hard Disk Drive Structure

- 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 alluminium or alluminium alloy, which is coated with a magnetic substance.

Computer Fundamentals: Memory Technologies

Hard Disk Drive:(HDD)

- It contains a "circular platter"(s) made up of non-magnetic material like alluminum or alluminium alloy, which is coated with a magnetic substance.
- Each platter is divided into hundreds of concentric rings called as "tracks" and each track is divided into fixed size of blocks called as "sectors".
- Size of each sector on each track is same usually
  size of the sector = 512 bytes.
- Cylinder: A cylinder is any set of all of tracks of equal diameter in a hard disk drive. It can be visualized as a single, imaginary, circle that cuts through all of the platters (and both sides of each platter) in the drive.
- Seek Time: it is the time required for a disk controller to move head from its current position to desired cylinder.
- Rotational Latency: Once head moved at desired cylinder, time required to rotate the platter to get alligned with desired sector is called as rotational latency.



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 alluminium or alluminium alloy, which is coated with a magnetic substance.

- Coating of magentic 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 concetric 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.

Magnetic Disk : Hard Disk Drive Structure

- There is one conductiong coil reffered 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 alligned with the desired sector, and time required for this rotation is reffered as rotational latency.

- Access Time = Seek Time + Rotational Latency.

Input Output Devices

- Devices which are connected to the motherboard externally through ports reffered as **peripheral devices** or **peripherals**.

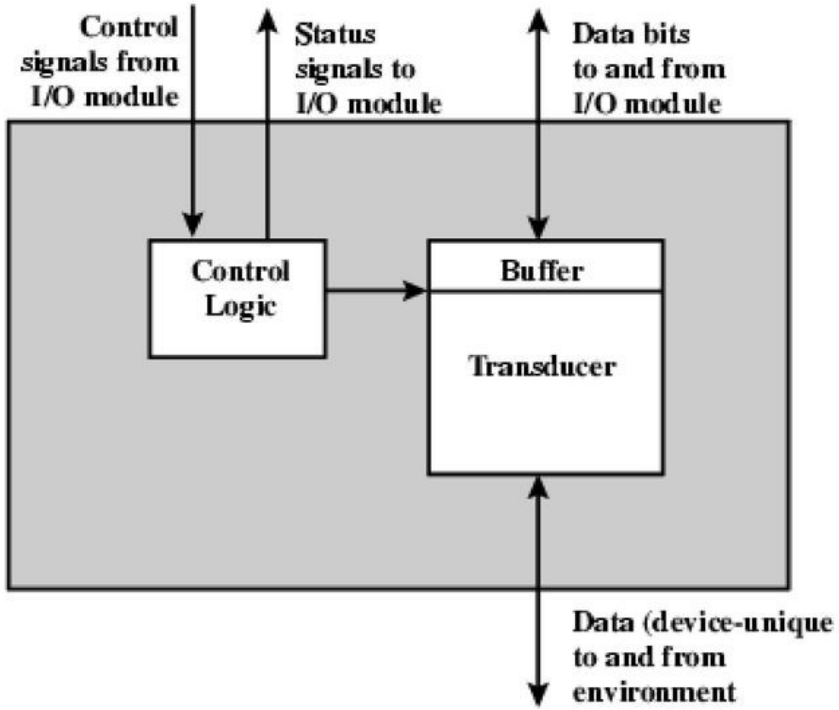
- An IO Devices are also reffered as an external devices.

 # 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...

Computer Fundamentals: IO Devices

External Device Block Diagram



Structure of an External Device:

- External Device has three major blocks:

1. Control Logic Block(Controller): controlls all the operations of that device.

2. Buffer: each device has gots its own memory in which data can be stored temporarily reffered 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 commuincation with the outside world.

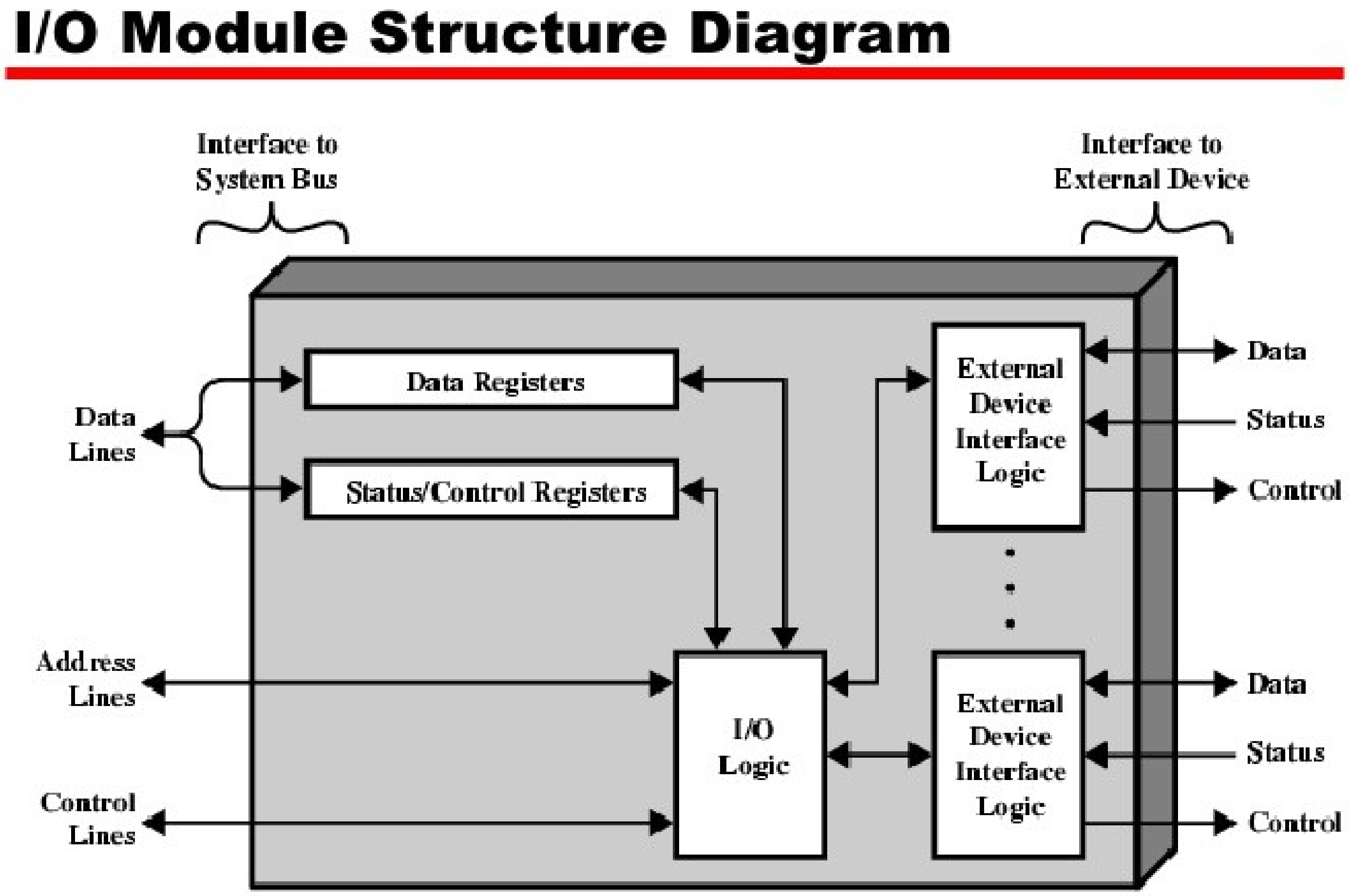
IO Modules/IO Ports:

- Core Computer system is not able to communicates directly with any external device and hence IO modules acts as an interface between core computer system and an io devices.

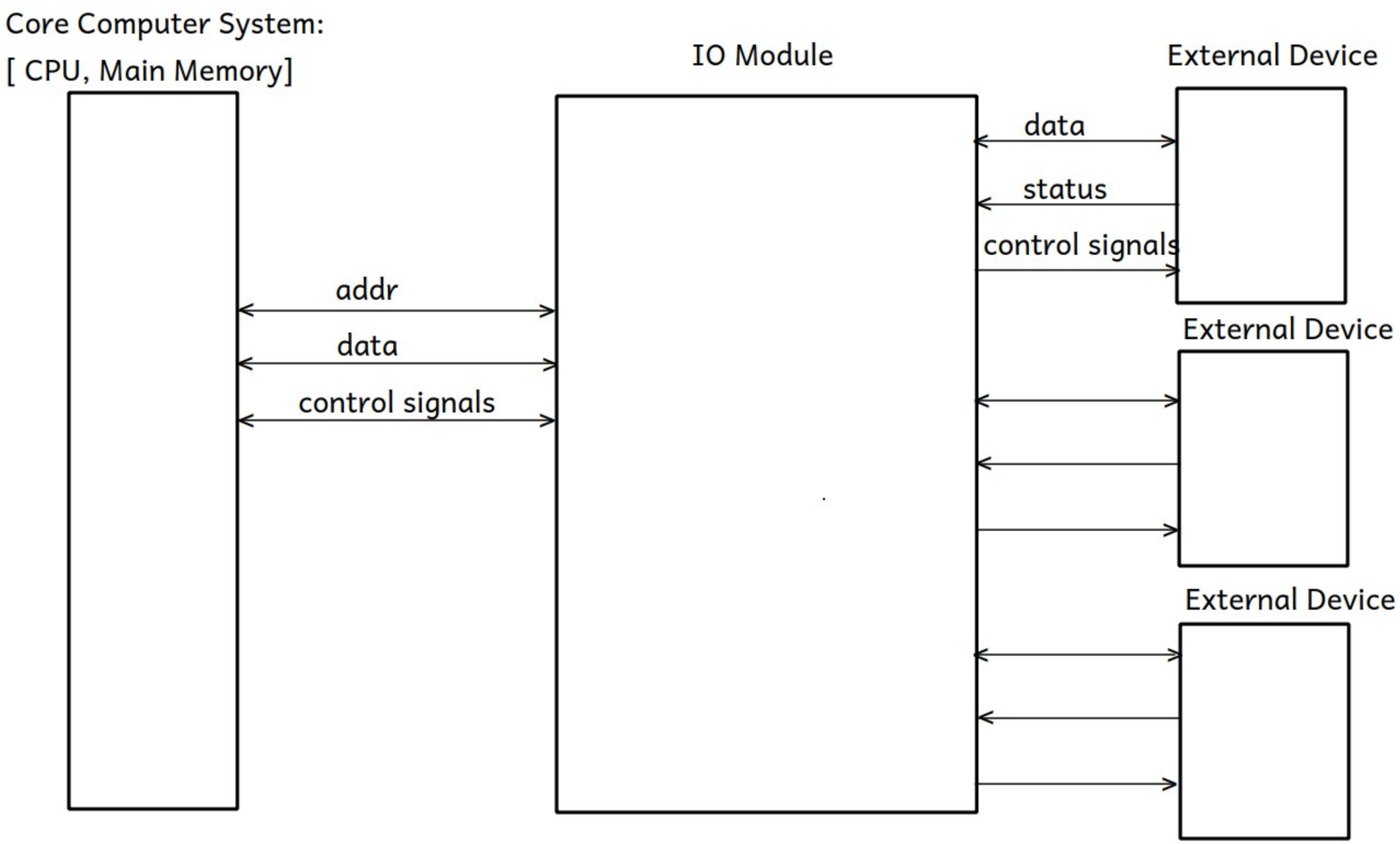
- IO Modules contains all the logic to communicates with an io devices.

- Single IO module can be used for communication between one device or with more one devices as well.

Computer Fundamentals: IO Devices



Computer Fundamentals: IO Devices



Computer Fundamentals: IO Techniques

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

- 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 IO is there into one program, and by means of executing that program by the CPU io can be done.

Advantages:

- Simple

Disadvantages:

- As the CPU remains wholely 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 recieved 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 unneccessary involvement of the CPU is there.

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.**

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.

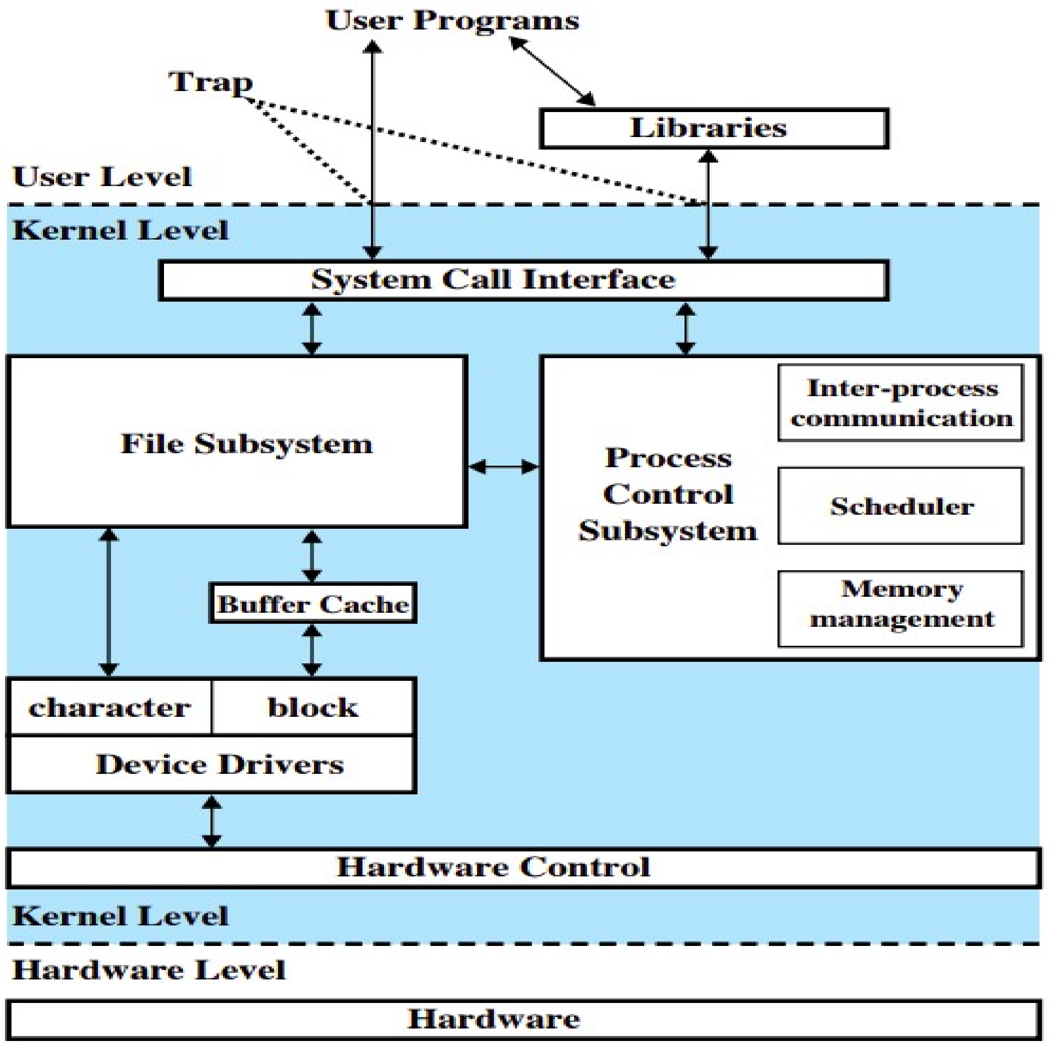
- It was first run on a machine **DEC-PDP-7** (Digital Equipment Corporation – Programmable Data Processing-7).

- 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 System Concepts



- Kernel acts as an interface between programs and hardware.
- Operating System has subsystems like **System Call Interface, File subsystem, Process Control Subsystem(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 reffered as a process.
- **File has space & Process has life.**

- From UNIX point of view all devices are considered as a file
- In UNIX, devices are catagorises into two catagories:

1. Character Devices: Devices from which data gets transfered character by character --> character special device file

e.g. keyboard, mouse, printer, monitor etc...

2. Block Devices: Devices from which data gets transfered block by block --> block special device file

e.g. all storage devices.

- **Device Driver:**It is a program/set of programs enable one or more hardware devices to communicate with the computer’s operating system.

- Hardware Control Layer/Block does communication with control logic block i.e. controller of a hardware.

 # 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 catagories of system calls:

1. Process Control System Calls: e.g. fork(), _exit(), wait() etc...

2. File Operations System Calls: e.g. open(), read(), write(), close() etc...

3. Device Control System Calls: e.g. open(), read(), write(), ioctl() etc...

4. Accounting Information System Calls: e.g. getpid(), getppid(), stat() etc...

5. Protection & Security System Calls: e.g. chmod(), chown() etc...

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

- In UNIX 64 system calls are there.

- In Linux more than 300 system calls are there

- In Windows more than 3000 system calls are there

- When system call gets called the CPU switched from user defined code to system defined code, and hence system calls are also called as **software interrupts/trap.**

Dual Mode Operation:

- System runs in two modes:

1. System Mode
2. User Mode

1. System Mode:

- When the CPU executes system defined code instructions, system runs in a system mode.
- System mode is also reffered as kernel mode/monitor mode/supervisor mode/priviledged mode.

2. User Mode:

- When the CPU executes user defined code instructions, system runs in a user mode.
- User mode is also reffered as non-priviledged mode.
- Throughout execution, the CPU keeps switch between kernel mode and user mode

Dual Mode Operation:

- Throughout an execution of any program, the CPU keeps switcesh in between kernel mode and user mode and hence system runs in two modes, it is reffered as **dual mode operation.**

- To differentiate between user mode and kernel mode one bit is there onto the CPU which is maintained by an OS, called as **mode bit**, by which the CPU identifies wheather currently executing instruction is of either system defined code instruction/s or user defined code instruction/s.

- In Kernel mode value of **mode bit = 0**, whereas

- In User mode **mode bit = 1.**

Process Management

- When we say an OS does process management it means an OS is responsible for process creation, to provide environment for an execution of a process, resource allocation, scheduling, resources management, inter process comminication, process coordination, and terminate the process.

Q. What is a Program?

- **User view:** Program is a set of instructions given to the machine to do specific task.

- **System view:** Program is an executable file divided into sections like exe header, bss section, data section, rodata section, code section, symbol table.

Q. What is a Process?

User view:

- Program in execution is called as a process.

- Running program is called as a process.

- When a program gets loaded into the main memory it is reffered as a process.

- Running instance of a program is reffered as a process.

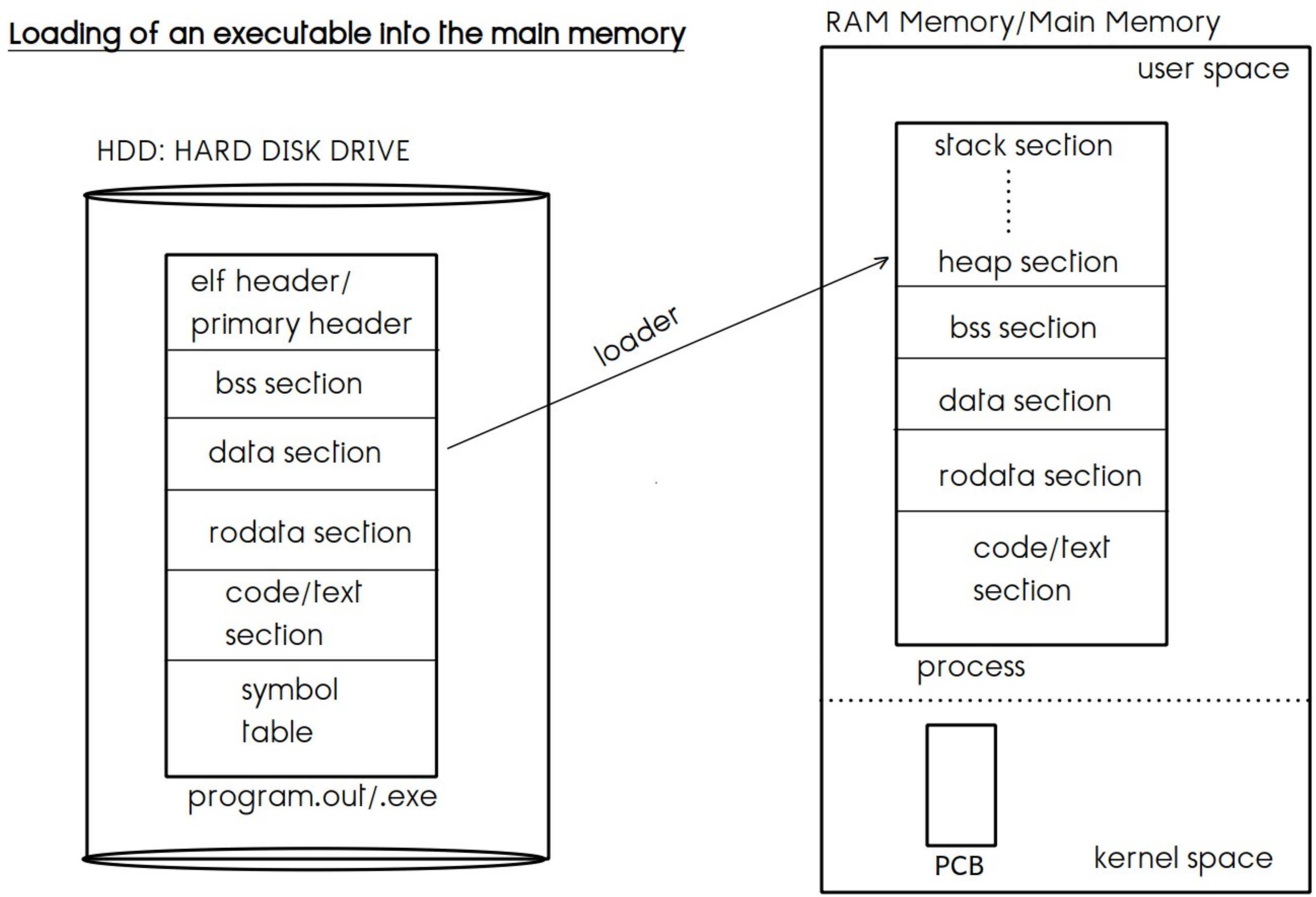
System view:

- Process is a file loaded into the main memory which has got bss section, rodata section, code section, and two new sections gets added for the process:

stack section: contains function activation records of called functions.

heap section: dynamically allocated memory

Operating System Concepts



- As a kernel, core program of an a OS runs continuosly into the main memory, **part of the main memory which is occupied by the kernel reffered as kernel space and whichever part is left is reffered as an user space**, so main memory is divided logically into two parts: **kernel space & user space.**

- User programs gets loaded into the user space only.

- When we execute a program or upon submission of a process very first one structure gets created into the main memory inside kernel space by an OS in which all the information which is required to control an execution of that process can be kept, this structure is reffered as a **PCB: Process Control Block**, is also called as a **Process Descriptor.**

- Per process one PCB gets created and PCB remains inside the main memory throughout an execution of a program, upon exit PCB gets destroyed from the main memory.

- PCB mainly contains: PID, PPID, PC, CPU sched information, memory management information, information about resources allocated for that process, execution context etc...

Process States:

- Throughout execution, process goes through different states out of which at a time it can be only in a one state.

- States of the process:

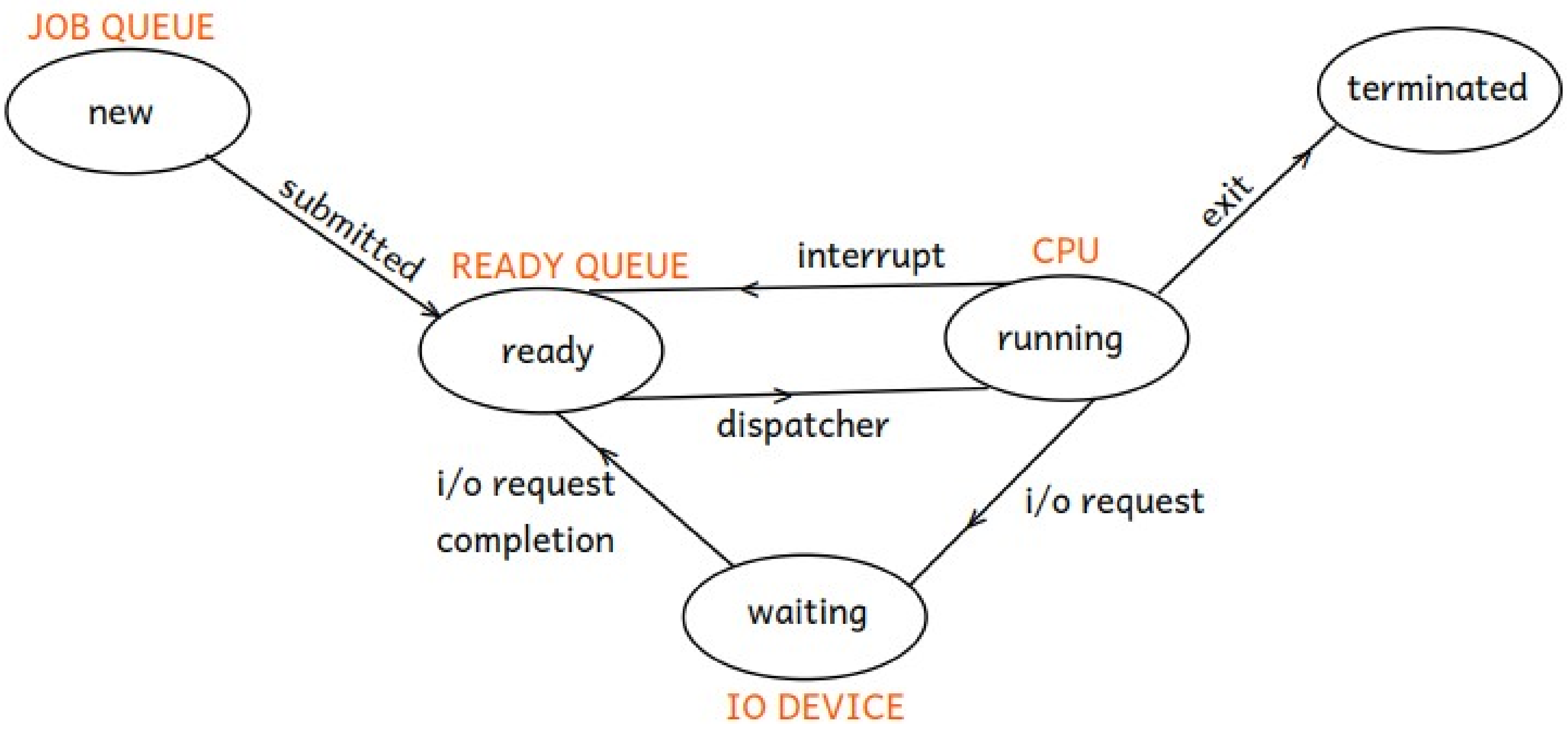
1. New state: upon submission or when a PCB for a process gets created into the main memory process is in a new state.

2. Ready state: after submission, if process is in a main memory and waiting for the CPU time.

3. Running state: if currently the CPU executing a process then it is reffered as in a running state.

4. Waiting state: if a process is requesting for i/o device then it is considered as that process is in a waiting state.

5. Terminated state: upon exit process goes into terminated state and its PCB gets destroyed from the main memory.



PROCESS STATE DIAGRAM

Features of an OS:

1. multi-programming: system in which more than one processes can be submitted/an execution of more than one programs can be started at a time.

- **degree of multi-programming:** no. of programs that can be submitted into the system at a time.

2. multi-tasking: system in which it seems that the CPU can execute more than one programs simultaneously/concurrently.

- time-sharing: system in which CPU time gets shared among all running programs.

3. multi-threading: system in which it seems that the CPU can execute more than one threads which are of either same process or are of different processes simultaneously/concurrently.

4. multi-processor: system can run on a machine in which more than one CPU's are connected in a closed circuit.

5. multi-user: system in which multiple users can loggedin at a time.

Process States:

- To keep track on all running programs an OS maintains few data structures reffered as kernel data structures:

1. Job queue: it contains list of PCB’s of all submitted processes.

2. Ready queue: it contains list of PCB’s of processes which are in a main memory and waiting for the CPU time.

3. Waiting queue: list of PCB’s of processes which are requesting for that particular device.

1. Job Scheduler/Long Term Scheduler: it is a system program which selects/schedules jobs/processes from job queue to load them onto the ready queue.

2. CPU Scheduler/Short Term Scheduler: it is a system program which selects/schedules job/process from ready queue to load it onto the CPU.

- Dispatcher: it is a system program which loads a process onto the CPU which is scheduled by the CPU scheduler, and the time required for the dispatcher to stops an execution of one process and to starts an execution of another process is reffered as **dispatcher latency.**

Context Switch:

- As during context-switch the CPU gets switched from an execution context of one process onto an execution context of another process, and hence it is reffered as "context-switch".

- context-switch = state-save + state-restore

- **state-save** of suspended process can be done i.e. an execution context of suspended process gets saved into its PCB.

- state-restore of a process which is scheduled by the CPU scheduler can be done by the dispacther, dispather copies an execution context of process scheduled by the cpu scheduler from its PCB and restore it onto the CPU registers.

- When a high priority process arrived into the ready queue, low priority process gets suspended by means of sending an interrupt, and control of the CPU gets allocated to the high priority process, and its execution gets completed first, then low priority process can be resumed back, i.e. the CPU starts executing suspended process from the point at which it was suspended and onwards.

- CPU Scheduler gets called in the following four cases:

1. Running -> Terminated
2. Running -> Waiting
3. Running -> Ready
4. Waiting -> Ready

- There are two types of CPU scheduling:

1. Non-preemptive: under non-preemptive cpu scheduling process releases the control of the CPU by its own i.e. voluntarily.

e.g. in above case 1 & case 2

2. Preemptive: under preemptive scheduling control of the CPU taken away forcefully from the process.

e.g. in above case 3 & 4.

- Following algorithms used for CPU Scheduling:

1. FCFS (First Come First Served) CPU Scheduling
2. SJF (Shortest Job First) CPU Scheduling
3. Priority CPU Scheduling
4. Round Robin CPU Scheduling
5. Multilevel Queue CPU Scheduling
6. Multilevel Feedback Queue CPU Scheduling

- As multiple algorithms are there for CPU scheduling and hence there are certain criterias which algorithm is best suited in which situation and which one efficient are reffered as CPU scheduling criterias.

CPU Scheduling Criterias:

1. CPU Utilization: one need to select such an algorithm in which utilization of the CPU must be as maximum as a possible.

2. Throughput: total work done per unit time. One need to select such an a algorithm in which throughput must be as maximum possible.

3. Waiting Time: it is the toal amount of time spent by the process in a ready queue for waiting to get control of the CPU from its time of submission. One need to select such an algorithm in whih waiting time must be as minimum as possible.

4. Response Time: it is a time required for the process to get first response from the CPU from its time of submission.

5. Turn-Around-Time: it is the total amount of time required for the process to complete its execution from its time of submission.

- One need to select such an algorithm in which turn-around-time must be as minimum as possible.

- Turn-around-time is the sum of periods spent by the process on ready queue for waiting and onto the CPU for execution from its time of submission.

Execution Time: it is the total amount of time spent by the process onto the CPU to complete its execution.

CPU Burst Time: total no. of CPU cycles required for the process to complete its execution.

- **Turn-Around-Time = Waiting Time + Execution Time.**

1. FCFS: First Come First Served

- In this algorithm, process which arrived first into the ready queue gets the control of the CPU first i.e. control of the CPU gets allocated for processes as per their order of an arrival into the ready queue.

- This algorithm is simple to impelement and can be implemented by using fifo queue.

- It is a non-preemptive scheduling algorithm.

 # Convoy effect: due to an arrival of longer processes before shorter processes, shorter processes has to wait for long time and their average waiting time gets increases, which results into an increase of an average turn-around-time and hence overall system performance gets down.

2. SJF(Shortest Job First):

- In this algorithm, process which is having minimum CPU burst time gets the control of the CPU first.

- SJF algorithm ensures minimum waiting time.

- Under non-preemptive SJF, algorithm fails as the submission time of all processes are not same, and hence it can be implemented as preemptive as well.

- Non-preemptive SJF is also called as **SNTF(Shortest-Next-Time-First).**

- Preemptive SJF is also called as **SRTF(Shortest-Remaining-Time-First).**

3. Priority Scheduling

- In this algorithm, each process is having priority in its PCB and process which is having highest priority gets control of the CPU first.

- Minimum priority value indicates highest priority.

- This algorithm is purely preemptive.

- Due to the very low priority process may gets blocked into the ready queue and control of the CPU will never gets allocated for such a process, this situation is reffered as a **starvation** or **indifinite blocking.**

- **Ageing:** it is a technique in which, an OS gradually increments the priority of blocked process, i.e. priority of blocked process gets incremented after some fixed time interval by an OS, so that priority of blocked process becomes sufficient enough to get control of the CPU, and starvation can be avoided.

4. Round Robin Scheduling Algorithm

- In this algorithm, fixed **time slice** or **time quantum** gets decided in advanced, and at a time control of the CPU may remains allocated with any process maximum for decided time-slice, once the given time slice is finished process gets suspended and control of the CPU gets allocated to the next process for maximum the decided time slice and so on each process gets control of the CPU in a round robin manner.

- If any process completes its execution before allocated time slice then control of the CPU gets allocated for the next process as soon as it is completed for maximum utilization of the CPU.

- This algorithm is purely preemptive.

- This algorithm ensures minimum response time.

Inter Process Communication

- Processes running into the system can be divided into two catagories:

1. Independent Processes:

- Process which do not shares data with any other process reffered as an independent process.

- Process which do not affects or not gets affected by any other process reffered as an independent process.

2. Cooperative Processes:

- Process which shares data with any other process reffered as cooperative process.

- Process which affects or gets affected by any other process reffered as cooperative process.

Inter Process Communication

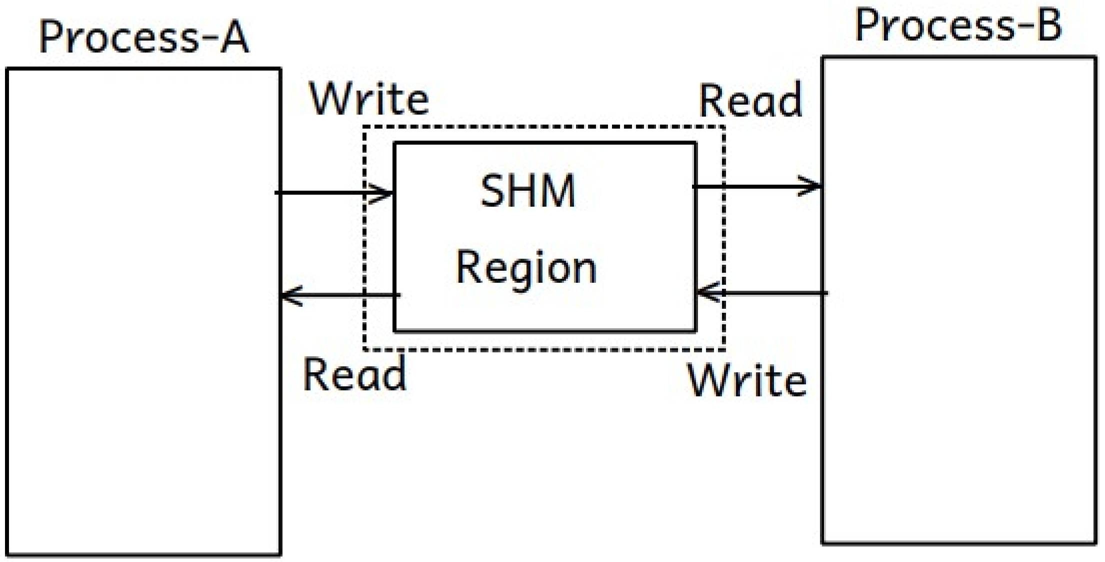
- Inter process communication takes place only between cooperative processes.

- There are two techniques by which IPC can be done/there are two IPC Models:

1. Shared Memory Model: under this technique processes can communicates with each other by means of reading and writing data into the **shared memory region** which is provided by an OS on request by a processes want to communicates.

2. Message Passing Model: under this technique, processes can communicates with each other by means of sending message.

- Shared Memory Model is faster than Message Passing Model.



SHARED MEMORY MODEL

Inter Process Communication

2. Message Passing Model: there are further different IPC techniues under message passing model.

i. Pipe:

- By using pipe mechanism one process can send message to another process, vice-versa is not possible and hence it is an **unidirectional communication** technique.

- By using Pipe only processes which are running on the same system can communicates.

- There are two types of pipes:

1. unnamed pipe: by using unnamed pipe, only **related processes** can communicates. e.g. pipe (|) command.

2. named pipe: by using named pipe **non-related processes** also can communicates. e.g. pipe() system call.

Inter Process Communication

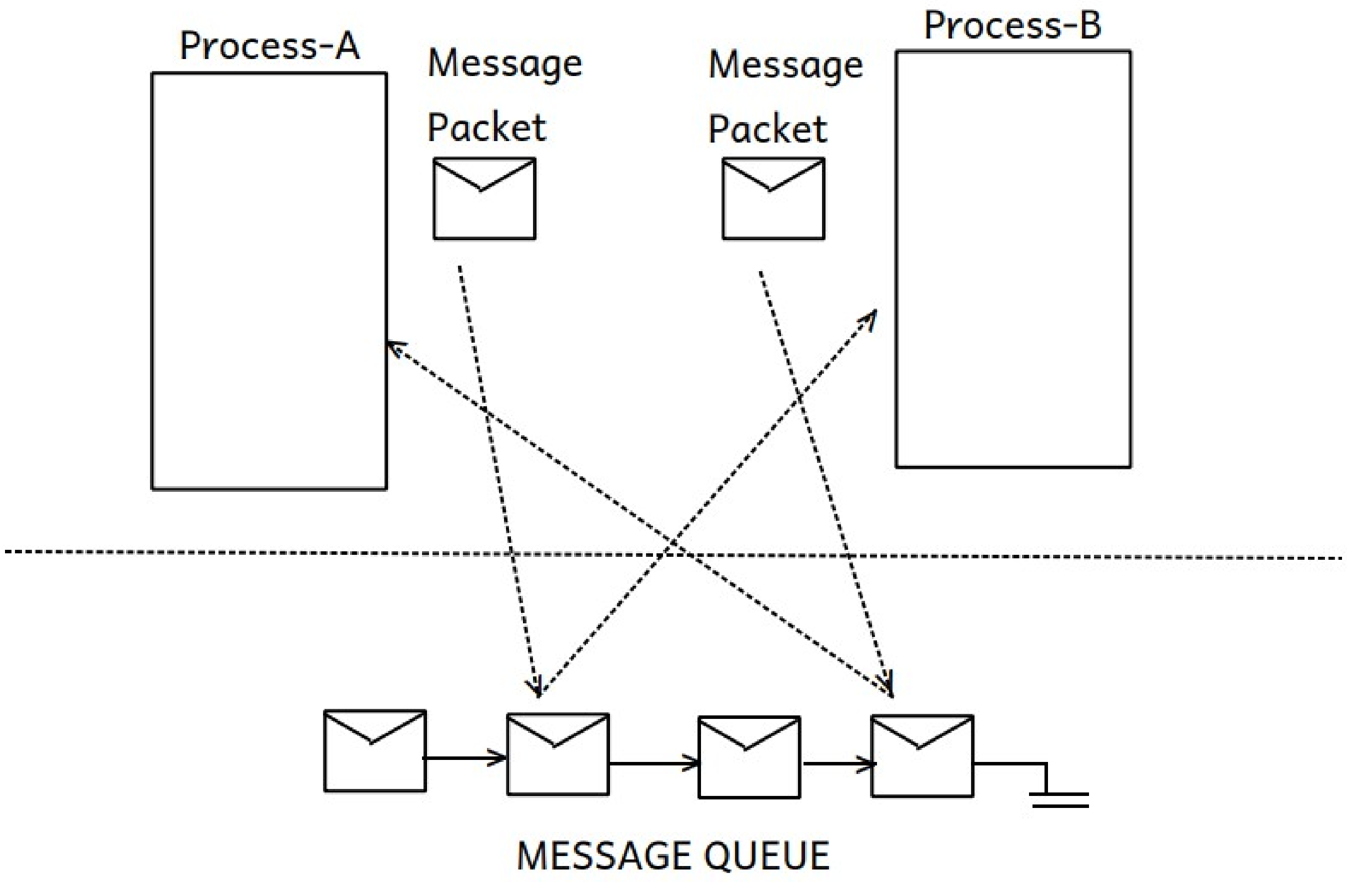
ii. Message Queue:

- By using message queue technique, processes can communicates by means of sending as well as recieving **message packets** to each other, and hence it is a **bidirectional communication**.

- **Message Packet: Message Header(Information about the message) + Actual Message.**

- Internally an OS maintains **message queue** in which message packets sent by one process are submitted and can be sent to reciever process and vice-versa.

Inter Process Communication



Inter Process Communication

iii. Signals:

- Processes communicates by means of passing signals as well.
- One process can send signal to another process.
- An OS send signal to any proces but any process cannot send signal to an OS.
- When we shutdown the machine an OS send SIGTERM signal to all processes due to which processes gets terminated normally, few processes can handle SIGTERM and even after recieving this signal from an OS continues execution, and hence to such processes an OS send SIGKILL signal due to which processes gets terminated forcefully.

e.g. SIGSTOP, SIGCONT, SIGSEGV etc...

Inter Process Communication

iv. Socket

- Limitation of above all IPC techniques is, only processes running on the same system can communicates, to overcome this limtation Socket IPC mechanism has been designed.

- By using socket IPC mechanism, process which is running on one machine can communicates with process running on another machine whereas machines are at remote distance from each other provided they connected in a network (either LAN/WAN/Internet).

- Socket = IP Address + Port Number.

e.g. chat application

Process Coordination/Process Synchronization

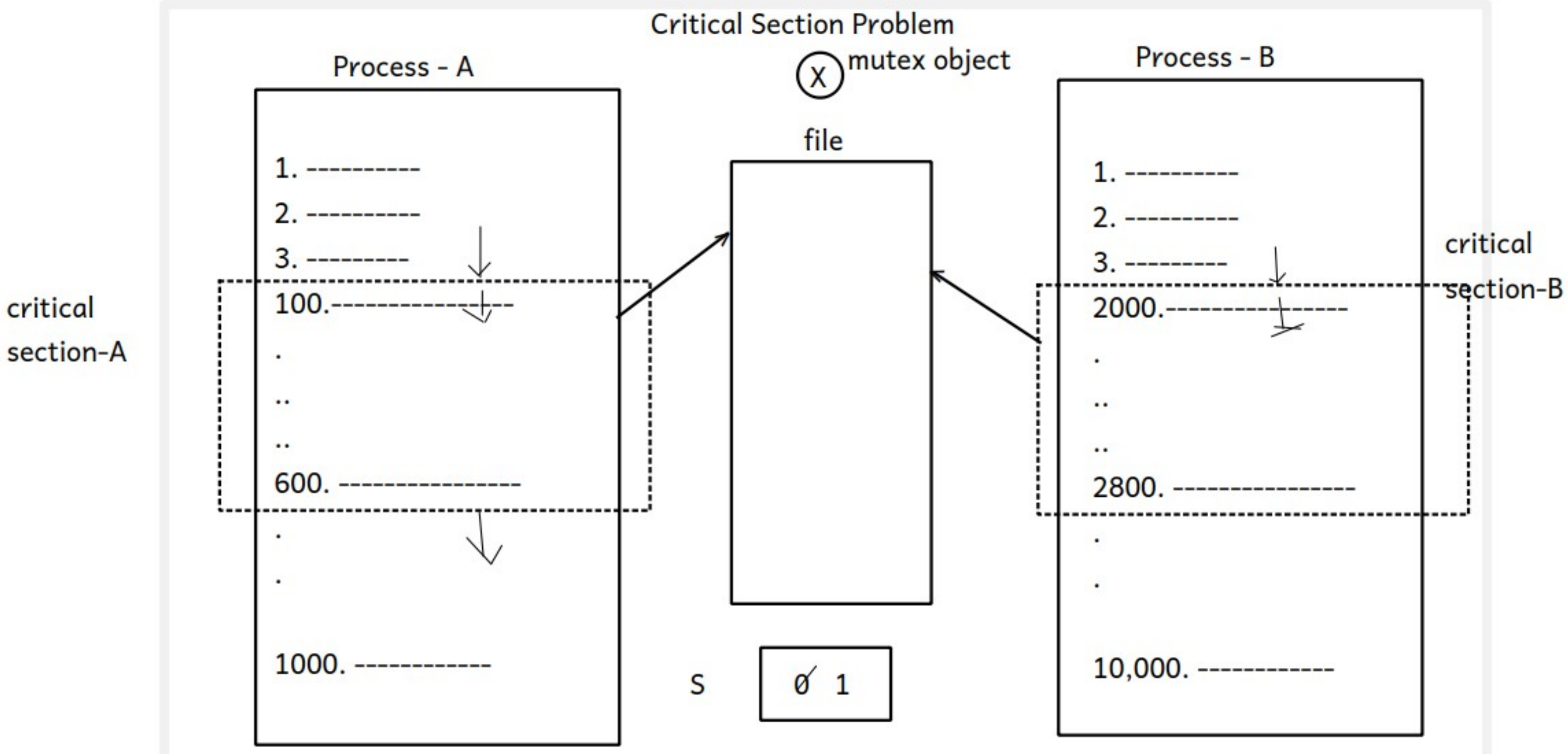
Race Condition: if one or more processes are trying to access a single resource at a time, race condition occures, and data inconsistency problem may takes place.

- Race condition can be avoided by an OS by

1. deciding their an order of allocation of resource for processes, and

2. whichever changes did by the last accessed process onto the resource remains final changes.

Critical Section Problem:



"data incosistency" problem occure in above case only when both the sections of process A & B are runniing at a same time, and hence these sections are reffered as critical section, and hence data inconsistency problem may occure when two or more processes are running in their critical sections at a same time, and this problem is also reffered as "critical section problem".

Synchronization Tools:

1. Semaphore: there are two types of semaphore

i. Binary semaphore: it can be used when at a time resource can be acquired by only one process.

- It is an integer variable having either value is 0 or 1.

ii. Classic semaphore: it can be used when at a time resource can be acquired by more than one processes.

2. Mutex Object: it can be used when at a time resource can be acquired by only one process.

- Mutex object has two states: locked & unlocked, and at a time it can be only in a one state either locked or unlocked.

Deadlock:

- There are four neccessary and sufficient conditions to occure deadlock.

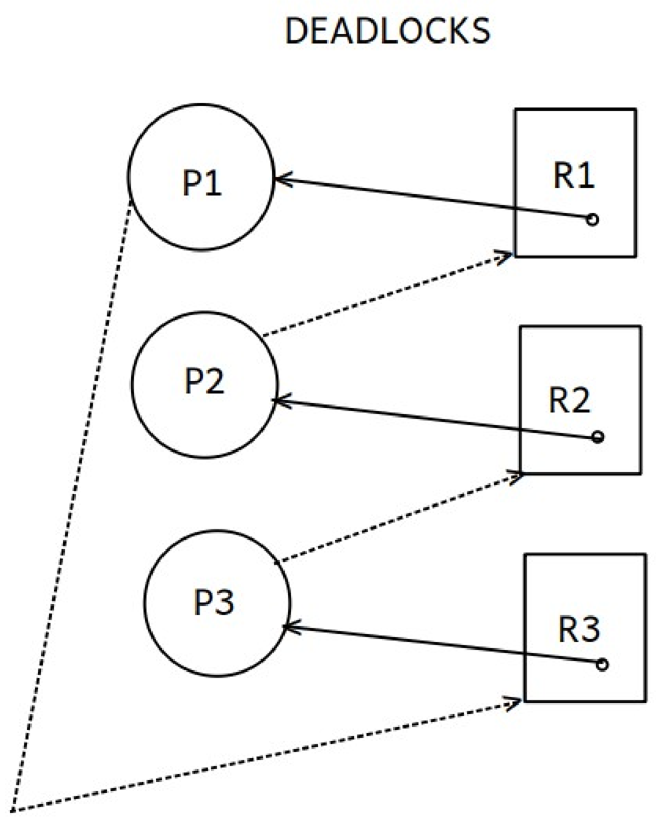
1. Mutual Exclusion: at a time resource can be acquired by only one process.

2. No Preemption: control of the resource cannot be taken away forcefully from any process.

3. Hold & Wait: every process is holding one resource and waiting for the resource which is held by another process.

4. Circular Wait: process P1 is holding resource and waiting for the resource held by another process P2, and process P2 is also holding resource and waiting for the resource held by process P1.

Deadlock: Resource Allocation Graph



- Three deadlock handling methods are there:

1. Deadlock Prevention: deadlock can be prevented by discarding any one condition out of four neccessary and sufficient conditions.

2. Deadlock Detection & Avoidance: before allocating resources for processes all the input given to deadlock detection algorithm in advanced and if there are chances to occure deadlock then it can be avoided by doing neccessary changes.

- There are two deadlock detection algorithms:

1. Resource Allocation Graph Algorithm
2. Banker’s Algorithm

3. Deadlock Recovery:

- System can be recovered from the deadlock by two ways:

1. Process termination: in this method any one process gets selected and terminated to recover system from deadlock, process which gets terminated is reffered as a **victim process.**

2. Resource preemption: in this method control of the resource taken away forcefully from a process to recover system from deadlock.