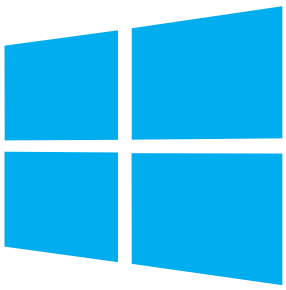
Operating System Notes



What is Operating system? explain the types of Operating system

- Operating system is a system software that manages the computer hardware
- It acts as intermediary between computer user and computer hardware
- e.g., windows, Linux, android, mac OS

- Types of OS
  - Batch OS -this type of OS does not interact with computer directly; it allows multiple users to use it at the same time ,E.g., of batch OS : payroll system, bank statements, etc.

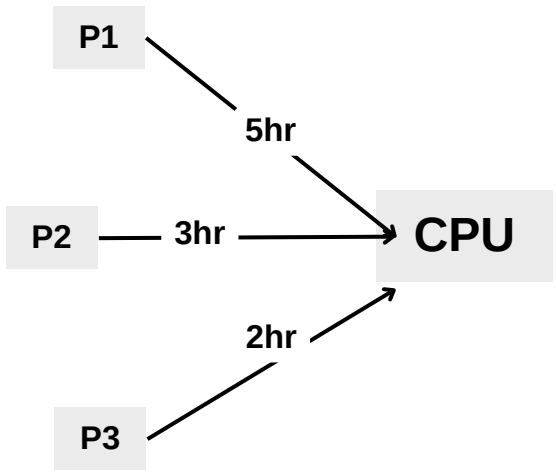
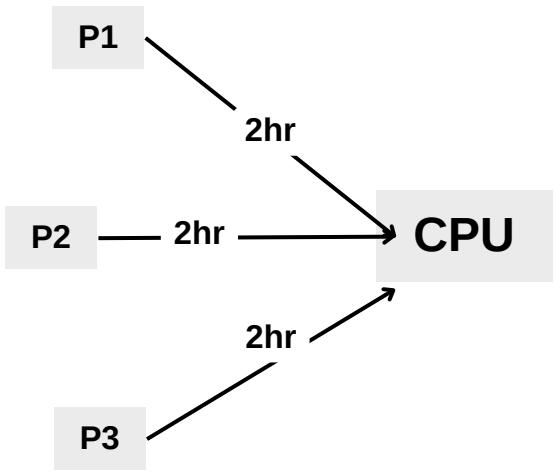
  - Time sharing - each process is given some time to be executed, the time given to each process to execute is called as quantum, after this time is over the OS switched over to next process

  - Real Time Sharing - these OS serves real time system, the time required to process is very small. E.g., air bags, missile system There are two types of Real time OS: hard(where time constraints are very strict) & soft(where time constraints are less strict)

  - Multi programming - there are one or more program loaded in the main memory which are ready to be executed, Here only one program can use the CPU at a time, while others have to wait

- Goals of OS
  - Convenience
  - Efficient
  - Both

Difference Between Multiprogramming and Multitasking

Multiprogramming	Multitasking
Multiprogramming is non-preemptive	Multitasking is Preemptive
CPU is allocated to the Process until the process is terminated or goes in waiting state	CPU is allocated for a time quantum
Here Execution of Program takes more time	Here Execution of program takes less time
CPU Idleness is more	CPU idleness is less
Waiting time and Response time is More	Waiting time and Response time is Less
	

What is System Call? Give some examples of System calls

- System call provides interface between a process and the operating system
- System call is mostly in the form of assembly language

- Types of system calls
  - Process control : end, abort, execute, create process, load process, terminate process
  - File management : create file, delete file, open, close, read, write
  - Device management : request/release device, read/write
  - Information management : get/set time, set date, get/set system data
  - Communication management : create/delete communication, send/receive messages
  - Protection : provides mechanism for controlling access to the resources

Explain the Process of Booting and Re-Booting

Booting

- Starting of the computer or an embedded system is called as booting

- Steps in Booting
  - Switching on power supply
  - Loading Operating system in the computer’s main memory
  - Keeping all application in the state of readiness in case user needs

- First program of set of instructions what execute/run when the computer is booted is called as BIOS (Basic Input Output System)

- BIOS is a firmware

Rebooting

- Restarting the system when it is already running is called as rebooting

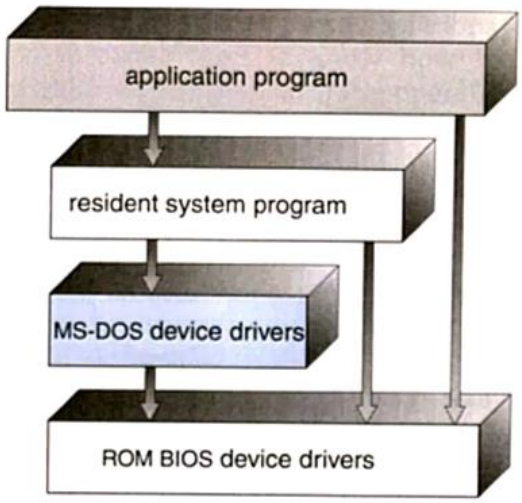
- When do we need rebooting
  - When software or hardware has been installed
  - When system is unusually slow

- Rebooting is faster than booting

Explain the Structure of OS

Simple OS structure

- Does not have a well-defined structure
- Ex: MS-DOS



Layered Approach

- Here the OS is divided into layers each on top of each other
- Bottom layer (layer 0) is hardware and topmost layer (layer n) is the user interface

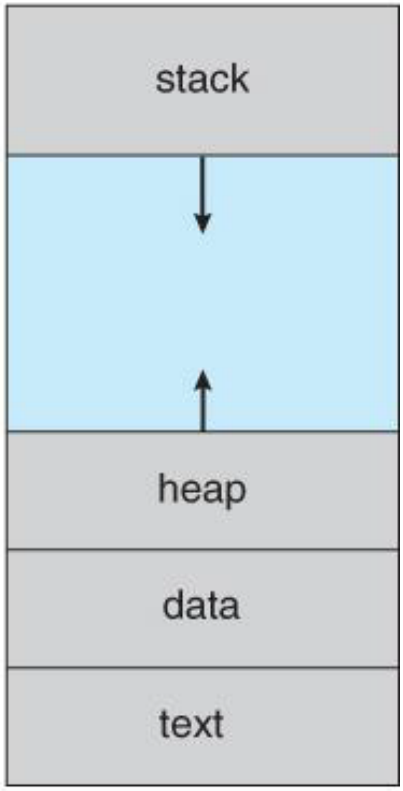
Layer 5	**User Program**
Layer 4	**I/O Management**
Layer 3	**Process Communication**
Layer 2	**Memory Management**
Layer 1	**CPU Scheduling**
Layer 0	**Hardware**

Explain What is Process Management

- Process management includes creating, running, terminating and assigning process to the CPU
- Process is program in execution
- Process is generated by OS and program is written by user
- When executable files get loaded in main memory and when it starts executing instructions one by one using process counter it is called as process
- We can run executable file by simply double clicking it or by running it on command prompt by inputting its name

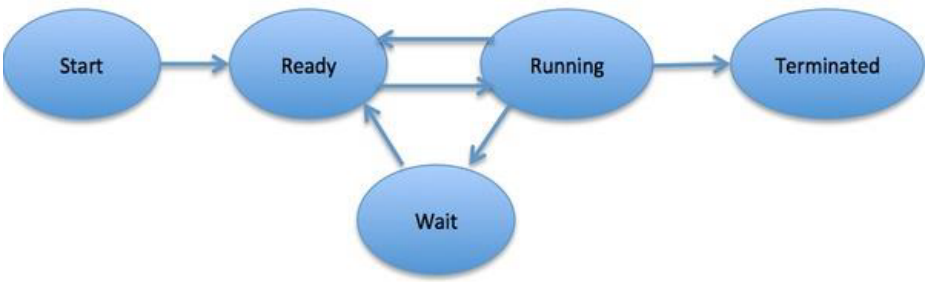
Structure of Process in Memory

- Stack is used to store temporary values
- These temporary values can be parameters or return values of functions or local variables
- Data section store global variables
- Heap section is used for dynamic memory allocation
- Text section carries the program code



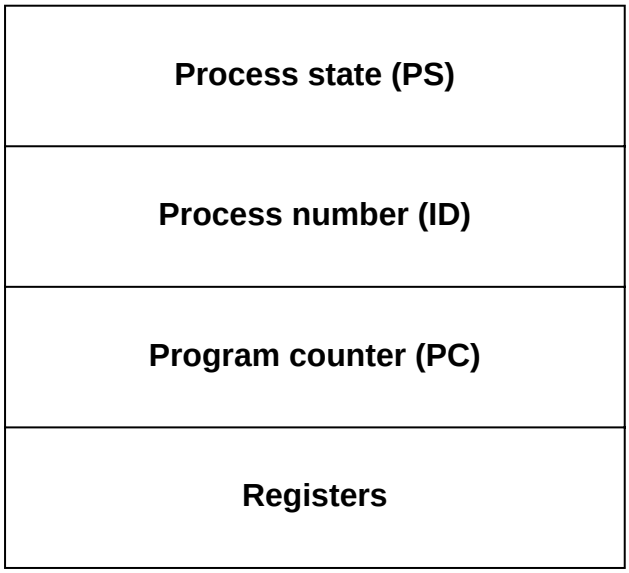
Explain Process State with Diagram

- New or start : the process is being created
- Running : the process is currently running in the CPU
- Waiting : the process is ready to execute but have to wait till the CPU is free or wait for I/O devices to be free
- Ready : the process is in main memory and has to wait to be assigned to the CPU
- Terminated : the process has been executed
- Blocked : if the process waits for to long then there is an additional state where it goes into deadlock and will never be able to execute again



Explain Process Control Block

- PCB is used to track the process execution status
- Process state : new, running, ready, waiting, terminated
- process number (ID) : it is the unique ID of the process
- program counter (PC) : it is used to indicate the address of next instruction which is going to be executed
- CPU register : accumulator, stack pointers, general purpose registers, index registers



What is Thread? Explain Benefits of Multi-Threading

- Thread is the basic unit of CPU utilization
- Thread is associated with thread ID, program counter, register set and stack
- Generally, a normal process is single threaded but processes can be multi-threaded
- If the system is multi-threaded it is very powerful and can perform multiple tasks at a time

Benefits of Multiprogramming

- Responsiveness : it allows the process to execute or run continuously even If the part of it is blocked
- Resource sharing : thread belong to a particular process they can also use the resources allocated to the process, so many threads can share resources of the process they belong to
- Economy : Process creation is costly due to memory allocation and resource allocation, as thread shares all resources from its process it is economical
- Utilization of multiprocessor architecture

What is CPU Scheduling? or Explain the Criteria of CPU Scheduling? or State the parameters of CPU Scheduling

- CPU scheduling is the process of assigning different jobs to the CPU by creating a scheduling using scientific sequence

- Parameters used in scheduling criteria
  - CPU utilization
  - Throughput
  - Turnaround time
  - Waiting time
  - Response Time

- CPU utilization
  - It is important to keep CPU busy all the time so it gets fully utilized, CPU utilization is measured in percentage
  - In real time CPU utilization is in the range of 40% to 90%

- Throughput : Number of processes completed in a unit of time is called as throughput

- Turnaround time : It is the time required for the process to be executed

- Waiting time : The time spend in the waiting queue is known as waiting time

- Response time : The time between actual submission and first response

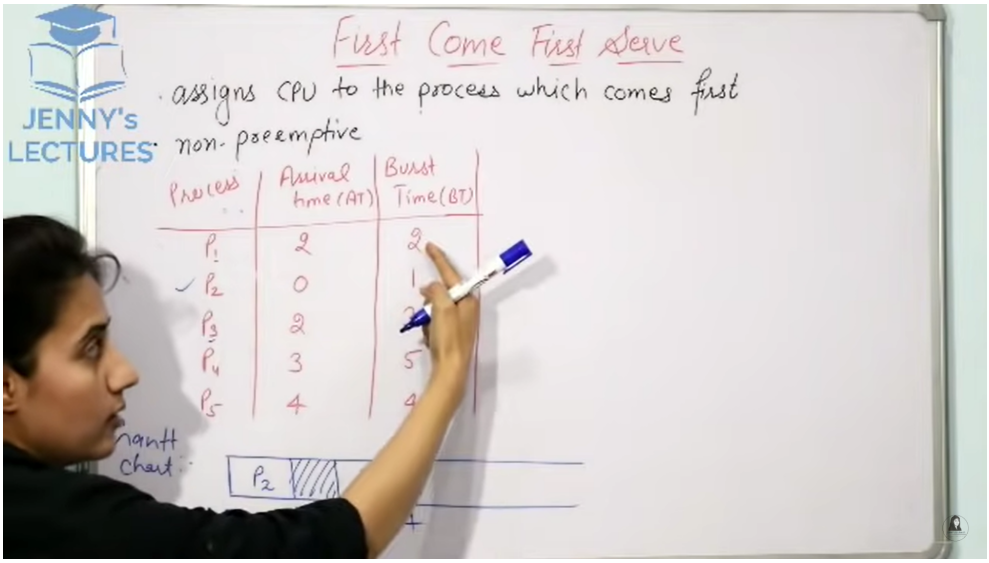
Difference Between Preemptive and Non-Preemptive

Preemptive	Non-Preemptive
CPU is allocated for a time quantum	CPU is allocated to the Process until the process is terminated or goes in waiting state
The Processes in Preemptive Scheduling can be interrupted	Processes cannot be interrupted
CPU utilization is higher as compared to non-preemptive scheduling	CPU utilization is less as compared to Preemptive Scheduling
Waiting time and Response time is Less	Waiting time and Response time is More
E.g., Windows 95	E.g., Windows 3

CPU Scheduling Algorithms

Check all the CPU Scheduling Algorithms by clicking on the Youtube links provided below

https://www.youtube.com/@JennyslecturesCSIT



- FCFS Algorithm
- SJF Algorithm
- SRTF Algorithm
- RR Algorithm

NOTE : CPU Scheduling Hand Written Algorithms will be provided soon

First Come First Serve

* Algorithm

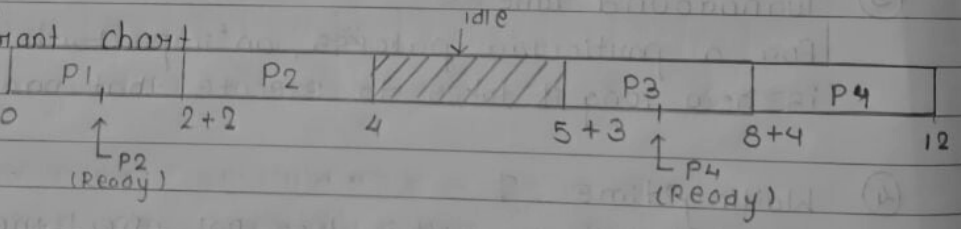
1] First come First serve

- Also known as First In, First Out
- Non-Preemptive

Execution

Process No	Arrival Time	Burst Time	CT	TAT	WT	RT
P1	0	2	2	2	0	0
P2	1	2	4	3	1	1
P3	5	3	8	3	0	0
P4	6	4	12	6	2	2

Grant chart



TAT = CT - AT

WT = TAT - BT

If Algorithm is Non-Preemptive

RT = WT

Average TAT = $\frac{2+3+3+6}{4} = \frac{14}{4} = 3.5$

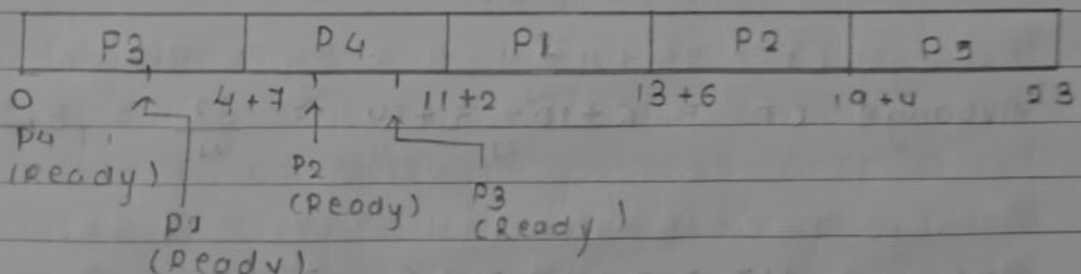
First Come First Serve (Page 2)

Average WT = $\frac{0+1+0+2}{4} = \frac{3}{4} = 0.75$

Average CT = $\frac{2+4+8+12}{4} = \frac{26}{4} = 6.5$

E.g.

Process No.	AT	BT	CT	TAT	WT	RT
P_1	2	2	13	11	9	9
P_2	5	6	19	14	8	8
P_3	0	4	4	4	0	0
P_4	0	7	11	11	4	4
P 5	7	4	23	16	12	12



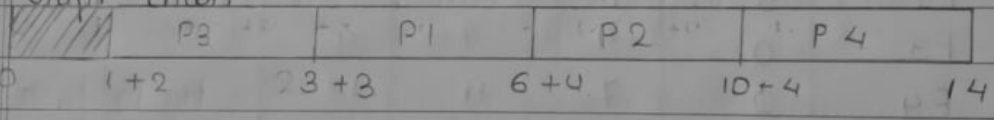
Shortest Job First (SJF)

2] Shortest Job First (SJF)

• Non - Preemptive

Process No	Arrival Time	Burst Time	CT	TAT	WT	RT
P1	1	3	6	5	2	2
P2	2	4	10	8	4	4
P3	1	2	3	2	0	0
P4	4	4	14	10	6	6

Gantt chart



Average TAT = $\frac{5+8+2+10}{4} = \frac{25}{4} = 6.25$

Average CT = $\frac{6+10+3+14}{4} = \frac{33}{4} = 8.25$

Average WT = $\frac{2+4+0+6}{4} = \frac{12}{4} = 3$

Shortest Remaining Time First (SRTF)

3] Shortest Remaining Time First (SRTF)

- Preemptive

Process No	Arrival time	Burst time	CT	TAT	WT	RT
P1	0	~~5 4 3 2 1~~ 0	9	9	4	0
P2	1	~~3 2 1~~ 0	4	3	0	0
P3	2	~~4 3 2 1~~ 0	13	11	7	7
P4	4	1	5	1	0	0

Gant chart:

P1	P2	P2	P2	P4	P1	P1	P1	P1	P3	P3	P3	P3
0+1	1+1	2+1	3+1	4+1	5+1	6+1	7+1	8+1	9+1	10+1	11+1	12+1

- RT = First time ~~arrival~~ allocated CPU - AT

Average CT = $\frac{9+4+13+5}{4} = \frac{31}{4} = 7.75$

Average TAT = $\frac{9+3+11+1}{4} = \frac{24}{4} = 6$

Average WT = $\frac{4+0+7+0}{4} = \frac{11}{4} = 0.75$

Average RT = $\frac{0+0+7+0}{4} = \frac{7}{4} = 1.75$

Priority Scheduling

4] Priority Scheduling Algorithm.

• Preemptive

Priority	Process NO	Arrival Time	Burst Time	CT	TAT	WT	RT
10	P1	0	5	15	15	10	0
20	P2	1	4	11	10	6	0
30	P3	2	2	4	2	0	0
40	P4	4	4	8	4	0	0

Higher the no. higher priority
Smaller the no. higher priority

Gant chart

P1	P2	P3	P3	P4	P4	P4	P4	P2	P2	P2	P1
0+1	1+1	2+1	3+1	4+1	5+1	6+1	7+1	8+1	9+1	10+1	11+4	15

Average CT = $\frac{15 + 11 + 4 + 8}{4} = \frac{38}{4} = 9.5$

Average TAT = $\frac{15 + 10 + 2 + 4}{4} = \frac{31}{4} = 7.75$

Average WT = $\frac{10 + 6 + 0 + 0}{4} = \frac{16}{4} = 4$

Round Robin Agorithm

5] Round Robin scheduling Algorithm (Time quantum decided) (2)

• Preemptive

Process No	Arrival Time	Burst Time	CT	TAT	WT	RT
P1	0	~~5~~ 3+0	13	13	8	0
P2	1	~~3~~ +0	8	7	4	1
P3	2	~~1~~ 0	5	3	2	2
P4	3	2 0	10	7	5	5
P5	4	~~3~~ +0	14	10	7	6

Ready Queue

P1	P2	P3	P4	P5	P1	P5

Gant chart

P1	P2	P3	P1	P2	P4	P5	P1	P5

0+2 2+2 4+1 5+2 7+1 8+2 10+2 12+1 13+1 14

Average CT = $\frac{13+8+5+10+4}{5} = \frac{40}{5} = 8$

Average TAT = $\frac{13+7+3+7+10}{5} = \frac{40}{5} = 8$

Average WT = $\frac{8+4+2+5+7}{5} = \frac{26}{5} = 5.2$

Average RT = $\frac{0+1+2+5+6}{5} = \frac{14}{5} = 2.8$

What does memory management do? and Explain Logical and Physical address

- Memory management in OS keeps track of the memory which is currently used and by whom it is used
- These facilities to know how much memory is fetch or occupied by the process
- Decides which process should be loaded in the main memory when space is empty
- Logical address : the address produced by the CPU is called as the logical address
- Physical address : the address loaded in the memory register is called as physical address
- MMU (Memory management unit) : converts the logical address into physical address

Explain what is loading? also explain the types of loading

- Creation of program requires loading the program in main memory
- The work of loader is to place load module in main memory
- There are 3 types of Loading, Absolute, relocatable and dynamic

- Absolute
  - It loads the load module at a specific memory location, this location is usually absolute and does not change when the program is loaded in memory

- Relocatable
  - The load module is placed anywhere is the main memory, this is possible if the assembler or compiler produce the address that are relative to some point

- Dynamic
  - Loading the modules into main memory on demand is called as dynamic loading, the program is loaded into main memory and is executed
  - When this program needs to call modules or another program it will check if it is loaded or not, If the program is not loaded it is loaded

What is Swapping?

- in the operating system is a memory management scheme that temporarily swaps out an idle or blocked process from the main memory to secondary memory and vice versa
- Swapping ensures proper memory utilization and memory availability for those processes which are ready to be executed.
- Swap-out is a method of removing a process from RAM and adding it to the hard disk.
- Swap-in is a method of removing a program from a hard disk and putting it back into the main memory or RAM.
- Advantages of swapping
  - multiple process can run by CPU using swapping
  - process do not have to wait long for execution
  - Improve the degree of multi-programming.

Explain Memory Partitioning and Its types

- Memory partitioning is a technique that is used to divide memory into several small parts known as partitions

- Advantages of memory partitioning
  - Useful for multiprogramming environment
  - Easy to fetch data from partitions
  - Easy to access partitions

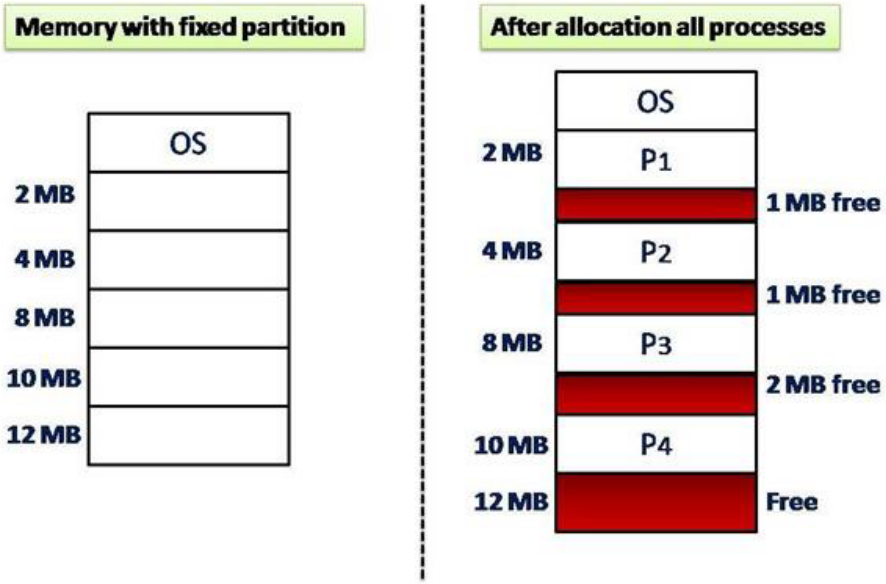
Static memory partitioning

- This method is also called as fixed partitioning
- The partitions are of fixed size
- If the process assigned to the memory partition is lesser than the fixed size then the remaining space cannot be used by any other process, This is called as internal fragmentation

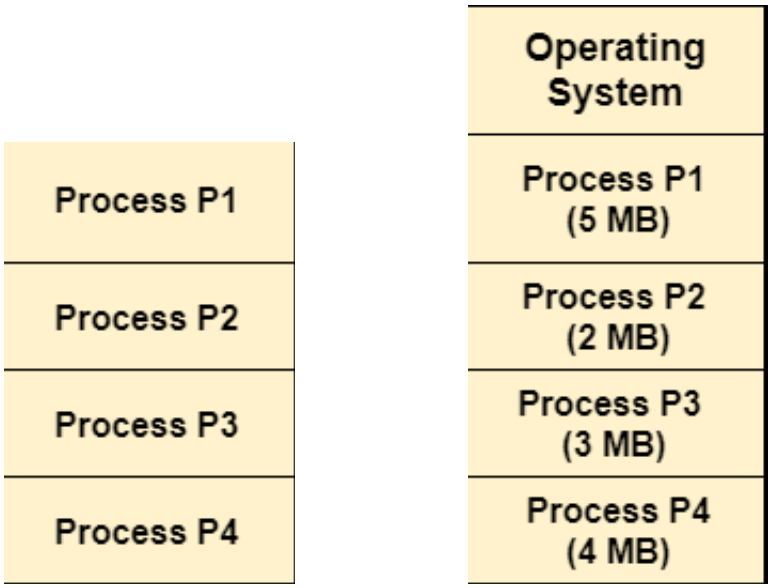
Dynamic memory Partitioning

- The partition is done as per the requirement of the process length or process size
- Maximum memory gets utilized
- When some process gets terminated its space is free but it might remain used by bigger process
- This is called as external fragmentation

Static memory partitioning

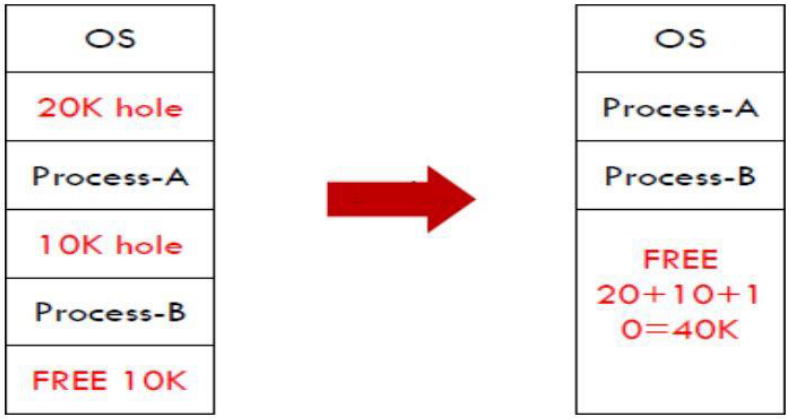


Dynamic memory Partitioning



Explain Compaction, Paging and Segmentation

- Compaction is a technique to overcome external fragmentation in dynamic memory partitioning
- All the free space is compacted together to form a new block or partition
- Compaction is a complex process



- Paging is another method which we can use instead of compaction to overcome external fragmentation in dynamic memory allocation
- - In paging physical memory is broken into fixed size blocks called as frames
- - And logical memory is broken into fixed size blocks called as pages
- - Size of one page is equal to the size of one frame

- In Operating System, Segmentation is a memory management technique in which the memory is divided into the variable size parts. Each part is known as a segment which can be allocated to a process.
- - Segment table is formed which contains segment number and it's offset

What is Deadlock?

- In multiprogramming environment process compete for physical as well as for logical recourses
- If a process P1 request for resource which is not available (held by some other process), P1 enters wait state
- It may happen that the wait state might never change because the resource requested by P1 is occupied by some other process
- This is called as Deadlock
- Ex : let’s assume two processes P1 and P2, P1 is holding tape drive and P2 is holding printer, deadlock can occur if P1 request for printer and P2 request for tape drive

List the Conditions for Deadlock

- Mutual exclusion
  - One process can hold only one resource at a time
  - Let’s say if P1 is holding onto a resource and if another process P2 request for the same resource at the same time then the process P2 has to wait until P1 releases this resource

- Hold and wait
  - P1 is hold a resource and is requesting for another resource which is held by another process (P2)

- No preemption
  - Resources cannot be taken away forcefully by any process
  - The process holding it should release it in order that other process can use it

- Circular wait
  - P1 is requesting for a resource which is held by P2 and P2 is requesting for resource which is held by P1

List the factors to Prevent Deadlock

- Elimination of Mutual exclusion
  - Deadlock can be prevented by eliminating mutual exclusion

- Eliminate hold and wait condition
  - We should ensure that the resource requested by the process is not held by any other process

- Removal of No preemption condition
  - Resources should not be preempted

- Elimination of circular wait
  - It is necessary to order the resources in such a way that circular waiting will never occur

- The technique to ensure that deadlock should not occur is called as deadlock avoidance
- A state can be called as safe if the system is capable of providing the resources to each and every process up to maximum value
- A state can be called as unsafe if the system is not capable of providing/ allocating resources to the process

What is a File? what does File management do?

- A file is a collection of data stored on mass storage
- E.g., in disk or on tape
- This data is divided into records (student information)
- Each record contains a number of fields (roll number, name)
- One or more field has a key field (roll number)

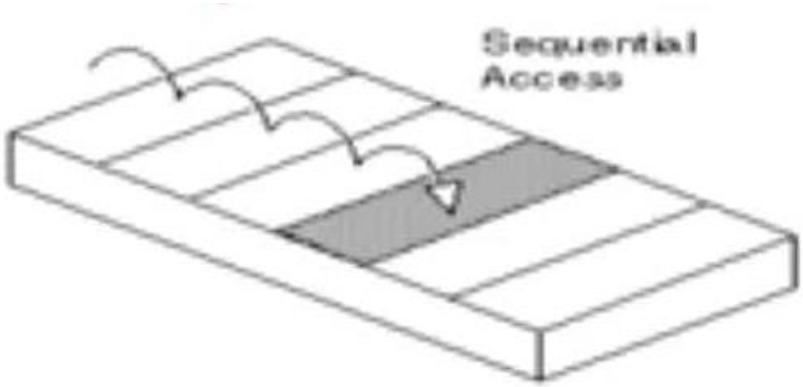
Work of File management

- To manage files and folders on hard disk
- Allocate space to file and folders
- De-Allocate space to file and folders
- Objectives of file management
  - Easy to fetch data
  - Data can be fetched faster
  - Operations like insert, delete, or update can be easy
  - reduces duplicate files
  - minimal cost of storage

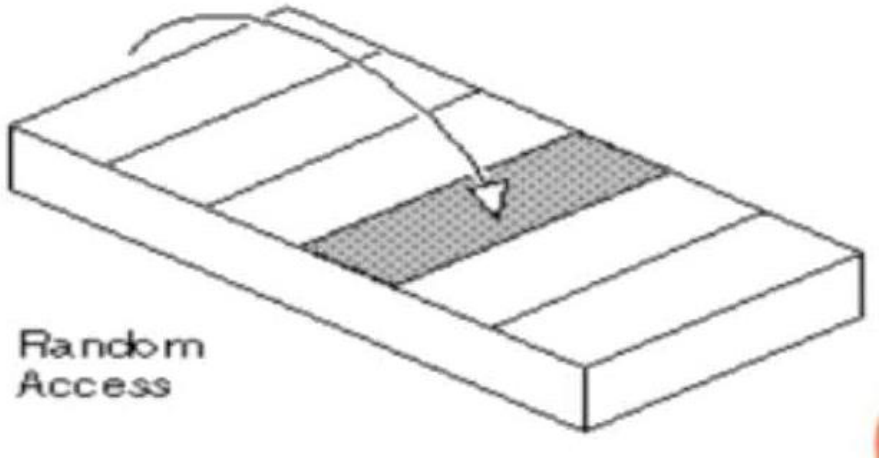
What is File Organization? or explain all the types of file organization types

- the technique to arrange the records on the secondary storage is called as file organization

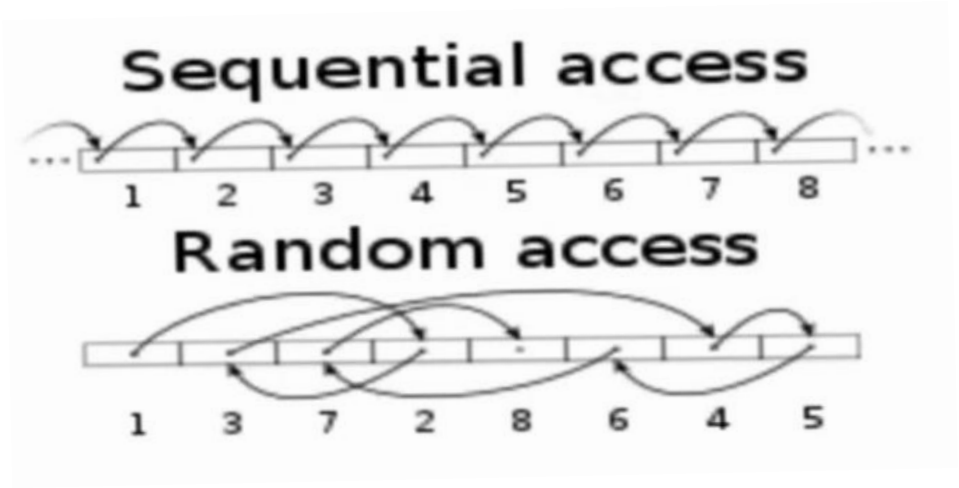
- Sequential Files
  - the records are stored in a sequence
  - sequence means the records are stored one after another
  - the records can be retrieved only in a sequence in which they are stored in
  - access of data is slow
  - if the last record has to be fetched all the records will be read before it



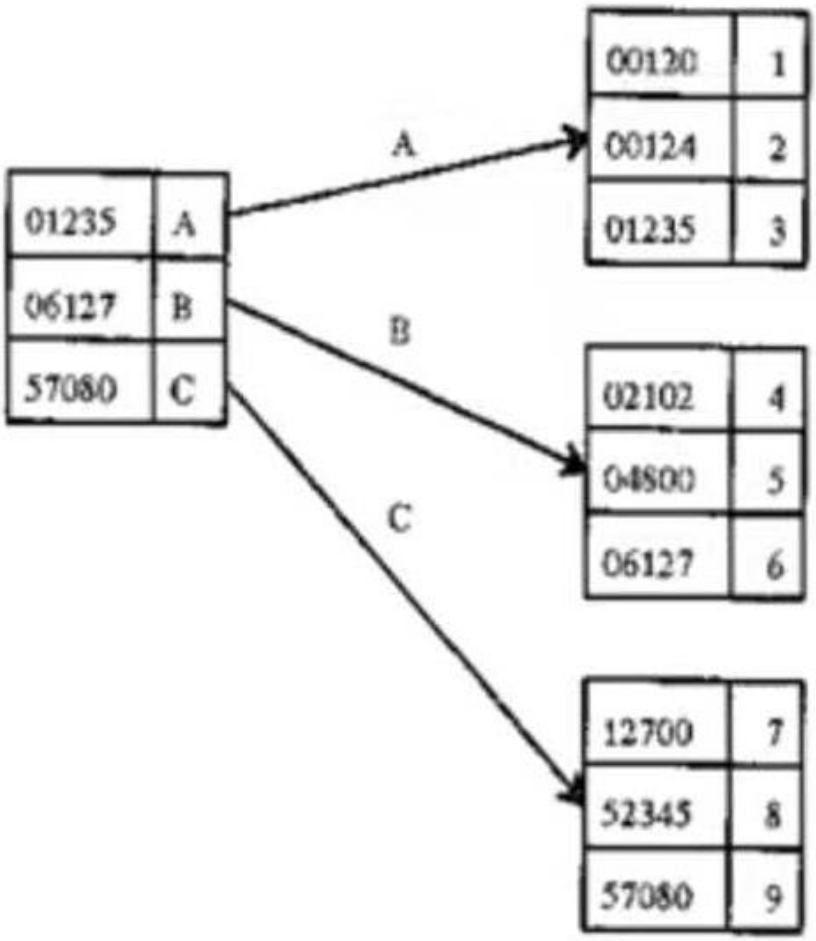
- Direct or random files
  - the records are not stored in a particular sequence
  - the key of the record is used to locate this file
  - each file is accessed directly without going through all previous files



- Random access is to store files in a random format



- Indexed Sequential method
  - The records are stored in an order (ascending or descending)
  - This order is based on the key
  - The index is stored in a separate file
  - This index contains the key values and the corresponding disk address
  - The index file is updated whenever new record is added or deleted from the file
  - This method is slower as compared to direct method
  - It also requires more space

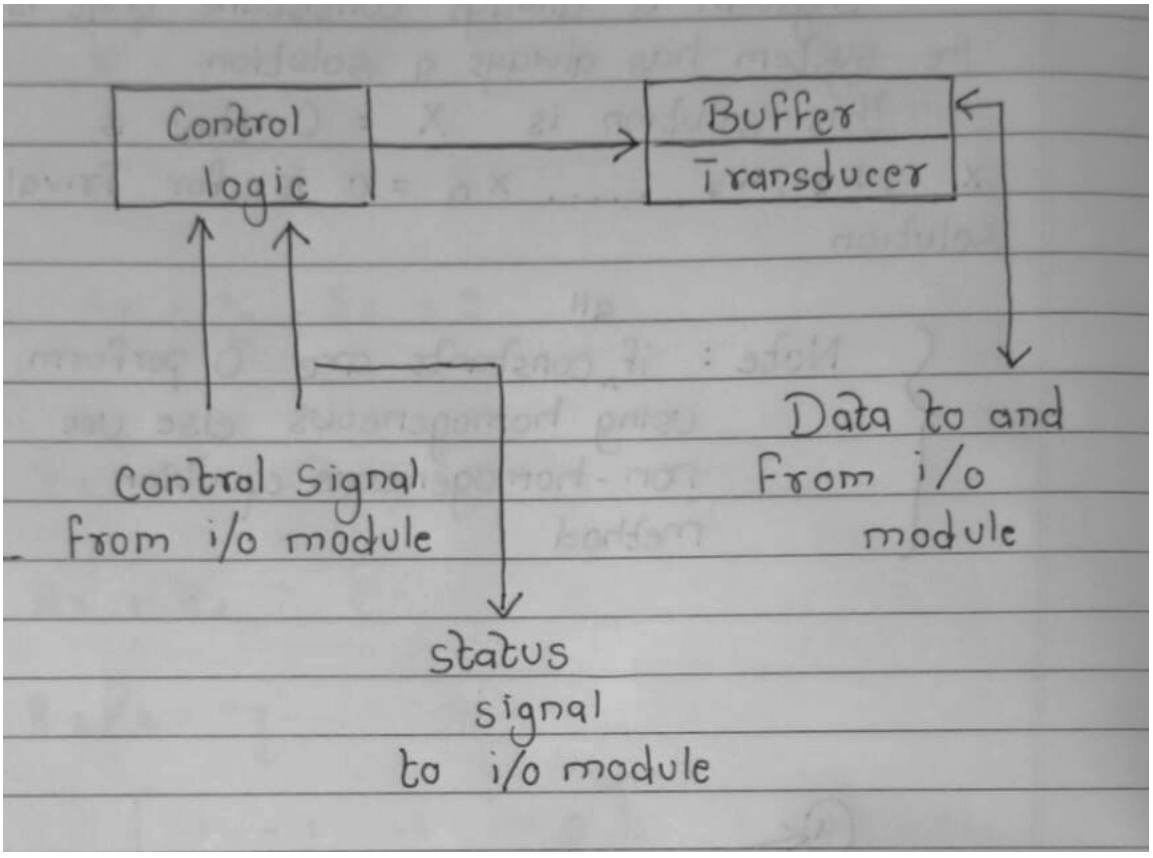


Input Output Devices

- To make a computer system interactive some input devices like keyboard, mouse, scanner, etc, and output devices like printer, plotter are attached to the system
- Input Output Devices can be grouped into the following
- Human readable : used to communicate with the user e.g., mouse, printer
- Machine readable : used to communicate with electronic equipment, eg. sensors, controllers
- Communication : used to communicate with network devices e.g, modems

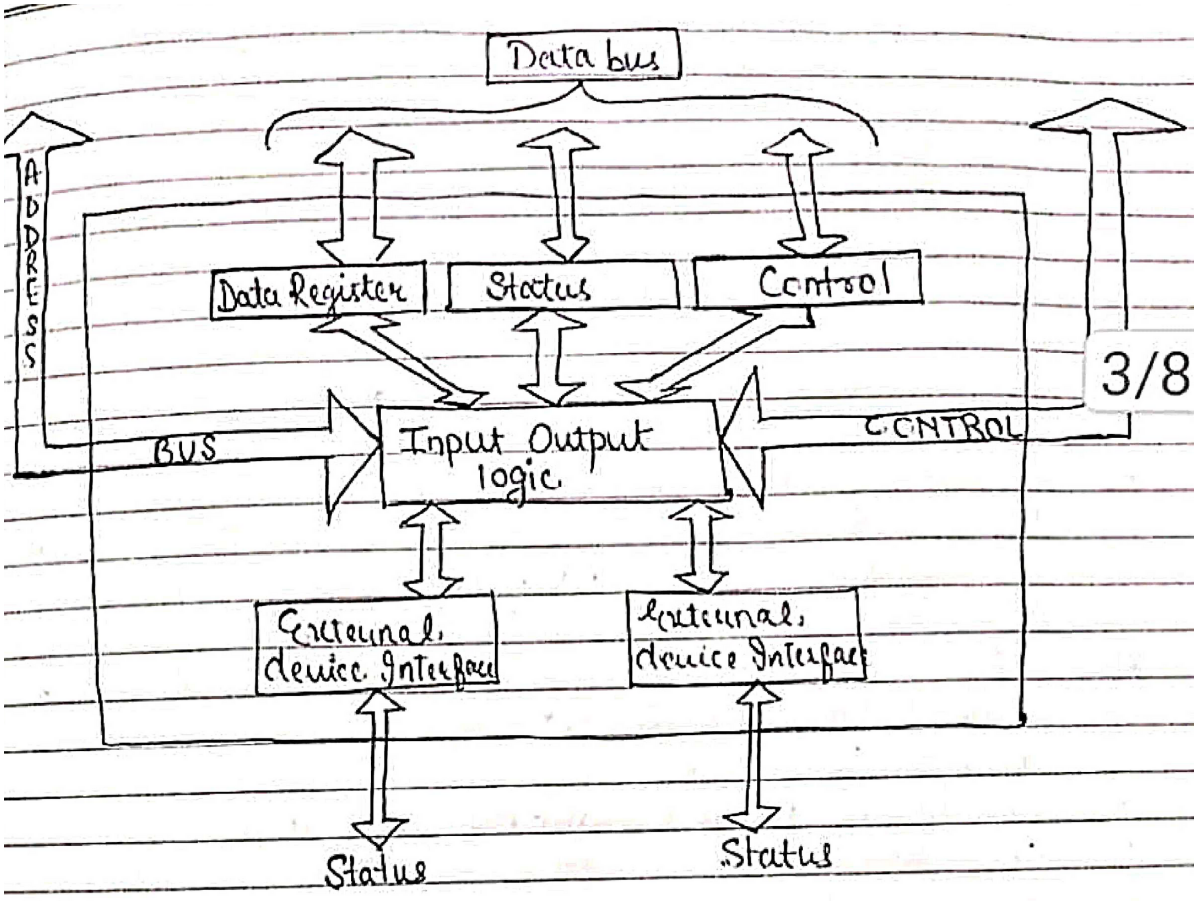
Short note on overview of I/O devices

- Input output devices can send and receive data to and from I/O module
- control signal sent by the I/O module decides if the data has to be read from the device or to write to the device
- Status Signal indicates if the device is read to transfer data or if it is busy with some work
- Control logic controls the operation of device in response to the direction form the I/O module
- Transducer converts electrical data to mechanical signal in case of output devices and mechanical to electrical in case of input devices
- Each Input Output device has a piece of memory called as buffer to temporary store data which is transferred



Structure of I/O module

- The I/O module connects to the rest of the computer through system bus
- Data Register : the data transferred to and from the module is temporary stored in one or more register
- Status register : status register provides status information
- Control register : it accepts detailed control information from the CPU
- I/O logic : uses control bus to interact with the CPU



Explain Programmed I/O or Polling

- Programmed I/O (Input/Output) is a basic I/O mechanism in computing where the CPU directly controls data transfer between peripheral devices and memory.
- In this method, the CPU initiates each data transfer operation by issuing specific commands to the I/O devices.
- It requires constant CPU involvement, making it inefficient for high-speed data transfer but simple to implement.

1. CPU issues Read command to I/O module
2. CPU read reads status register of I/O module
3. if I/O module is not ready repat step 2
4. Else read from I/O module
5. Write data into main memory

Explain Interrupt Driven I/O

- Interrupted I/O is an I/O mechanism in computing where the CPU doesn't continuously poll or control data transfer between peripherals and memory.
- Instead, it relies on hardware interrupts to signal when an I/O operation is complete. When an I/O device finishes its task, it sends an interrupt request to the CPU, which temporarily suspends its current task to service the interrupt.
- This approach allows the CPU to handle other tasks while waiting for I/O operations to complete, making it more efficient than programmed I/O for multitasking.

Direct Memory Access (DMA)

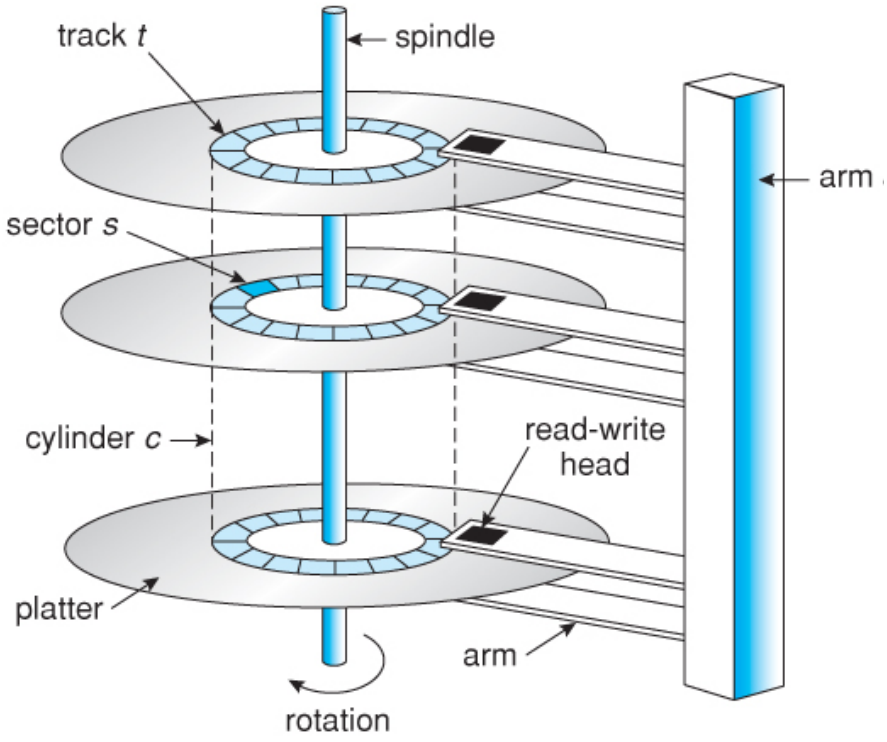
- Direct Memory Access (DMA) is an I/O mechanism in computing that enables high-speed data transfers between peripheral devices and memory without involving the CPU extensively.
- DMA controllers take over the data transfer process after the CPU initiates the operation, allowing the CPU to perform other tasks in the meantime.
- This reduces CPU overhead and improves overall system performance, making it especially useful for large data transfers or multitasking environments.

Why need Disk Scheduling?

- Disk scheduling is done by OS to schedule I/O request arriving for the disk. it is also known as I/O scheduling
- It is important because multiple I/O request may arrive by different processes and only one request can be served at a time by the disk
- thus other request have to wait and needed to be scheduled

Write short note on Disc

- Disk is divided into tracks, cylinders and sectors
- Hard disks drives are organized as a concentric stack of disk or platters
- Each platters has 2 surfaces and each surface has 2 read/ write heads
- each platter has the same no of tracks



- Track : Each disk is divided into a number of concentric circle, inner track are more densely packed with data
- Sector : All the tracks are sub divided into sectors
- Cylinder : A set of corresponding tracks in all the surface is called as cylinder

- The total number of bytes that can be stored in a data pack is = number of cylinders x tracks per cylinder x sectors per track x bytes per sector

Factors affecting the Speed of Disk

- Their are 2 components that affect the speed of disk that are Access time and Disk bandwidth

Access Time

- Access Time depends on 2 factors Seek time and Rotational Latency
- Seek Time : Seek time is the time taken to locate the disk arm to a specified location on the track
- The disk scheduling algorithm with least seek time is better
- Rotational Latency (RT) : It is the time taken by the disk sector to rotate into a position
- Access Time = Seek time + RT + disk bandwidth

Disk Bandwidth

- The bandwidth of the disk is the ration of total number of bytes transferred and total time for this transfer
- Bandwidth depends on the rotating speed

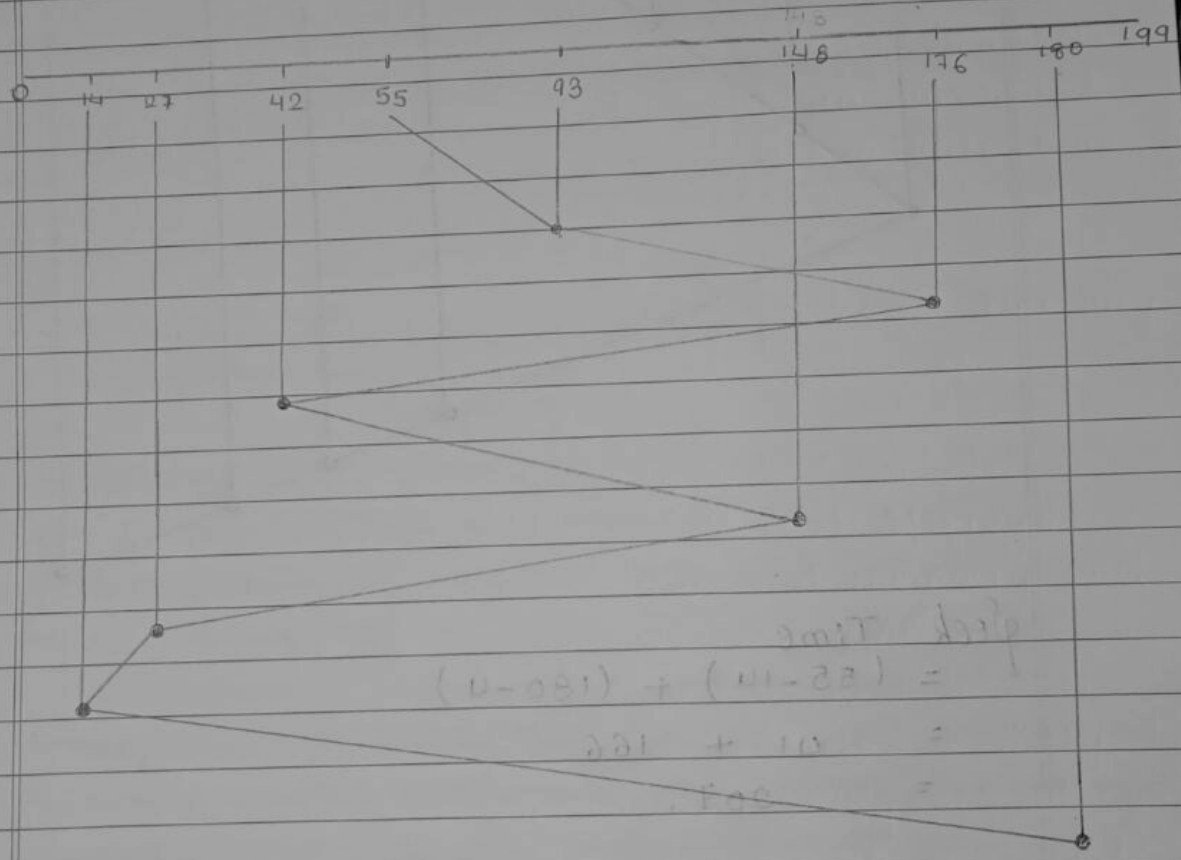
Disk Scheduling Algorithms

- FCFS : The request are addressed in the order they are arrived in the disk queue
- SSJF : The request with Shortest Seek time are executed first
- SCAN : In Scan algorithm the disk arm moves into a particular direction and serves all the request and after reaching the end of disk it changes the direction ad serves the request in this direction
- C-SCAN : Goes in one direction then to the end of another direction and then goes to the other end performing a circle
- LOOK : is similar to scan but instead of going to the end it goes to the last request and then reverses the direction from there serving all the request in path
- C-LOOK : C-LOOK is similar to C-Scan algorithm, the arm goes to the last request in one direction and from there it goes to the last request at the other end

FCFS (First Come First Serve)

1] First come first serve

Disk contain 200 track. Request Queue 93, 176, 42, 148, 27, 14, 180 current position of Arm is 55.



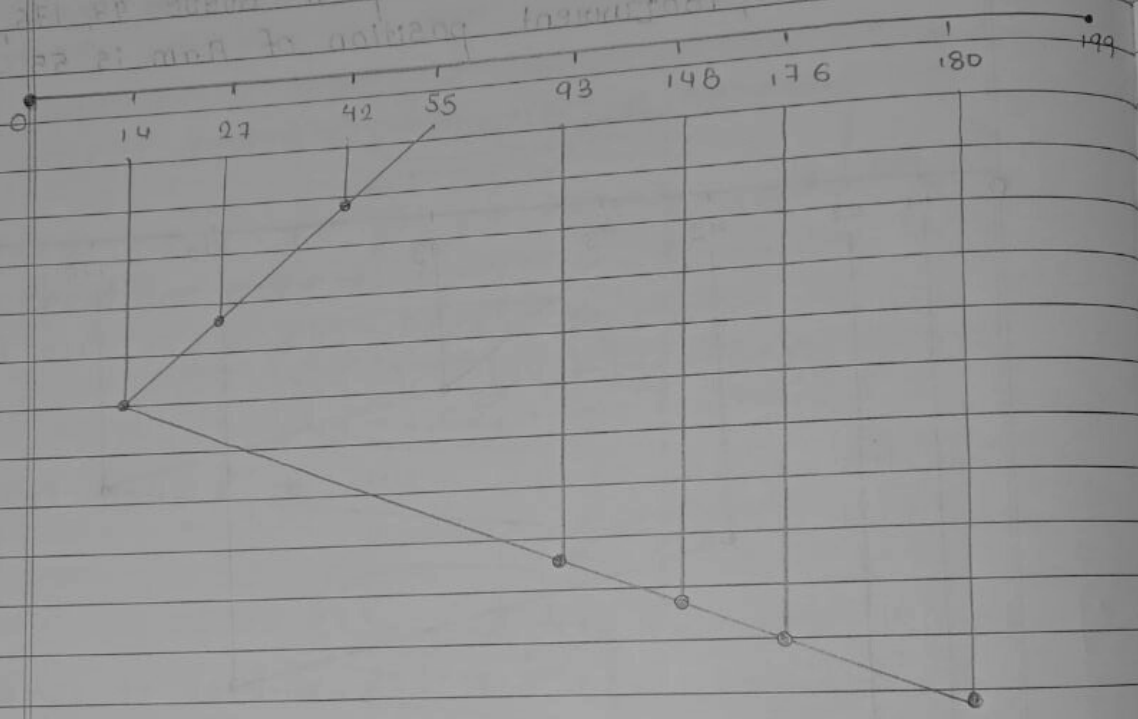
Seek Time

$= (93 - 55) + (176 - 93) + (176 - 42) + (148 - 42) + (148 - 27) + (27 - 14) + (180 - 140$

$= 661$

SSTF (Shortest Seek Time First)

2] Shortest Seek Time First

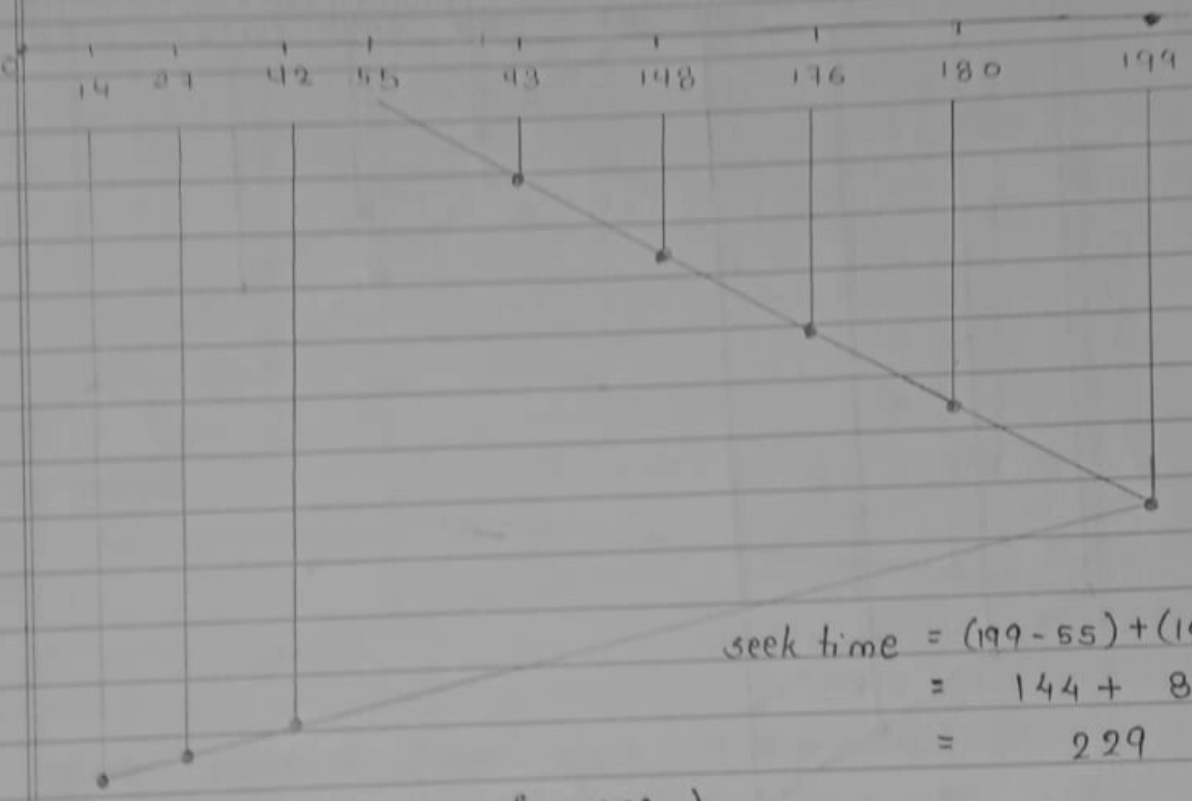


Seek Time
= (55 - 14) + (180 - 4)
= 41 + 166
= 207.

SCAN

3] qScan

(towards larger)

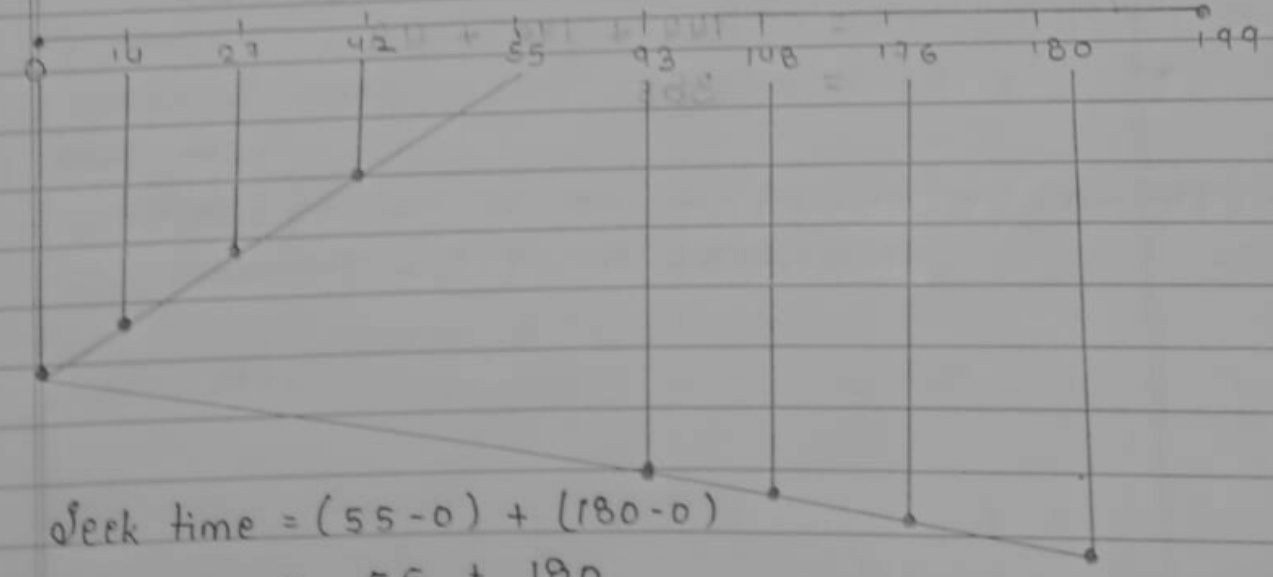


seek time = (199 - 55) + (199 - 14)

= 144 + 85

= 229

(towards qsmaller.)



Seek time = (55 - 0) + (180 - 0)

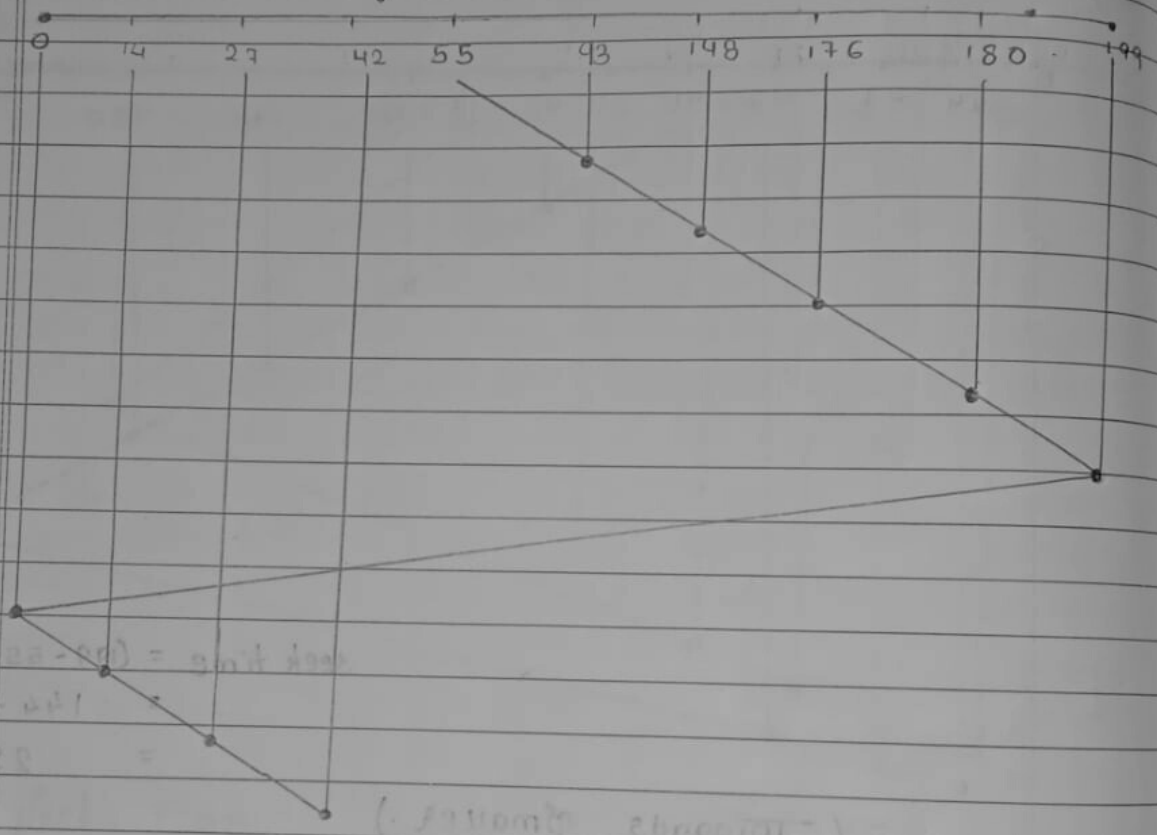
= 55 + 180

= 235

C - SCAN

4] C- Scan (Circular Scan)

(Towards Larger)



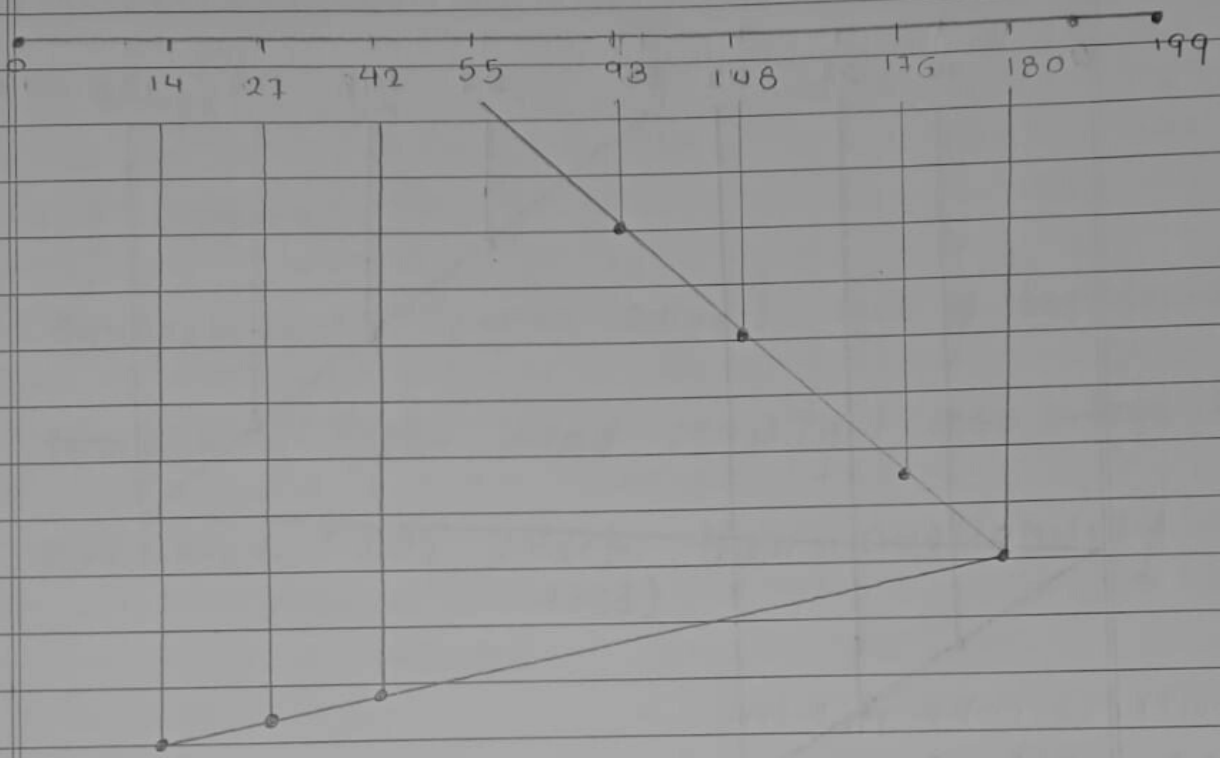
Seek Time = (199 - 55) + (199 - 0) + (42 - 0)

= 144 + 199 + 42

= 385

LOOK

5] Look



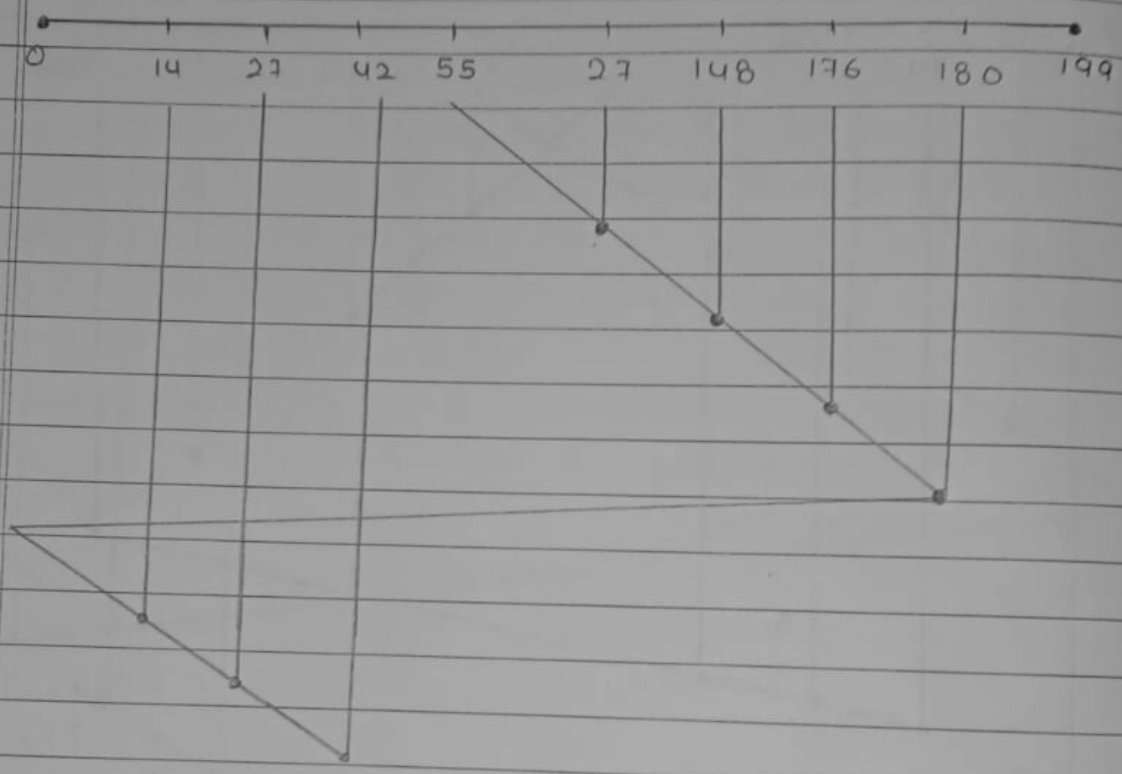
Seek Time = (180 - 55) + (180 - 14)

= 125 + 166

= 291

C - LOOK

6] C - Look



Seek Time = (180 - 55) + (180 - 0) + (42 - 0)
= 125 + 180 + 42
= 347

Goals of Protection

- Computer Consists of a Collection of objects, hardware and software
- Each Object has a unique name and can be accessed through a well defined set of operations
- Protection problem - ensure that each object is accessed correctly and only by those processes that are allowed to do so
- Need of protection
  - Prevent the mischievous and intentional violations
  - Improve reliability

Security Violation Categories

- Breach of confidentiality : unauthorized reading of data
- Breach of Integrity : unauthorized modification of data
- Breach of Availability : Unauthorized destruction of data
- Theft of service : Unauthorized use of resources
- Denial of service (DOS) : prevention of legitimate use

THE END