UNIT- 5

RAID :- (Redundant Arrays of Independent Disks)

RAID is a data storage virtualization technology that combines multiple Physical disk drive components into one or more logical units for the purpose of data redundancy performance improvement or both.

⇒ RAID is a method of storing data on multiple hard disk. When disk are arranged in a RAID configuration, the computer sees them all as one large disk.

Why Data Redundancy :- Data redundancy although takes extra space but adds to disk reliability. In case of failure of disk if the same data is also blocked up onto another disk, we can retrive the data is and go on with the operation.
On the other hand if the data is spread across just multiple disks without RAID technique the loss of a single disk can affect the entire data.

(1) **Reliability :-** How many disk faults can System tolerate.

(2) **Availability :-** How available is the system for actual use.

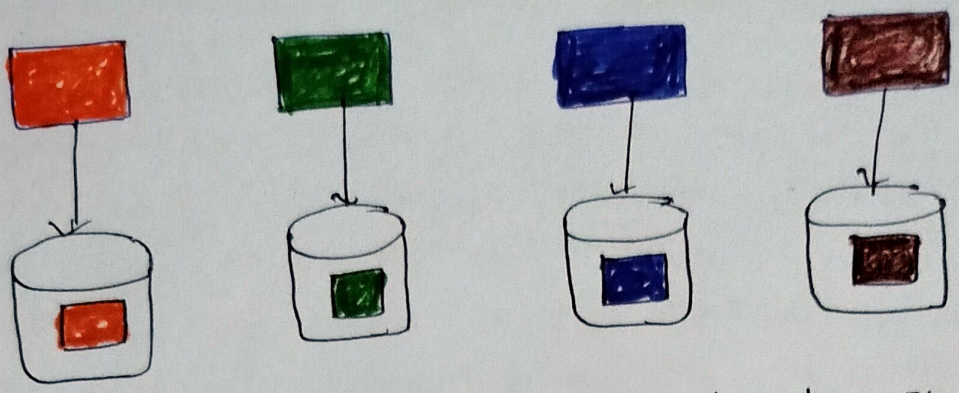
Performance!- How good is the response time How high is the throughput (rate of Processing work); where are other Parameters also.

Capacity!- given a set of N-disks each ~~time~~ with B blocks, how much useful capacity is available to the users.

⇒ RAID places data on multiple disk and allowing input/output operations to overlap in a ballenced way improving performance to overlap in a balanced way exiure of multiple disk increases the mean time extra. failure (MTBF) storing data redundancy.

RAID can have different redundancy levels achieving different performance and reliability a seven different RAID levels (0-6)

RAID-0! (Non redundant striping.
Data striped across all disks in an array No Parity disk.

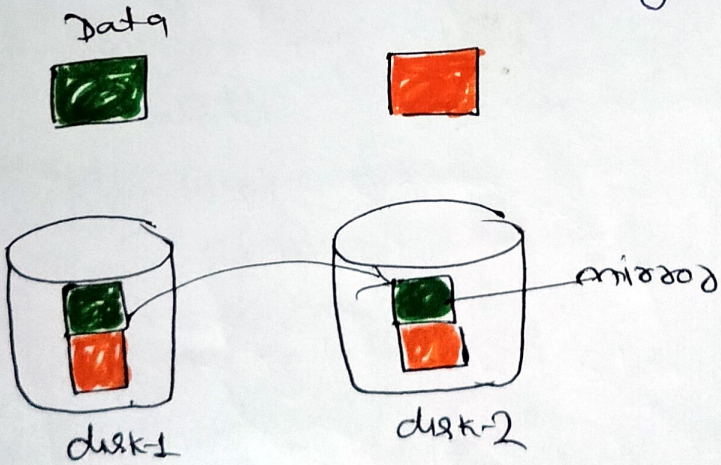


(RAID-0 non redundant striping)

Advantages:- good performance with N disks roughly N time speedup.

Disadvantages:- Poor reliability if one disk failure.

RAID – 1: (mirroring) keeps a mirrored (shadow) copy of data



Advantages:- good reliability one disk failure and good read performance.

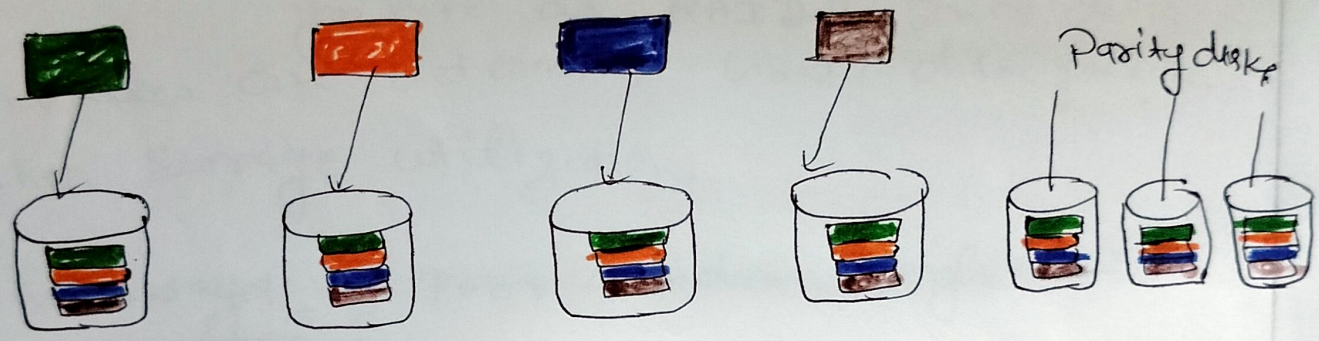
disadvantages: High cost done dat disk requires one parity disk.

RAID level-2 :- error-correction Parity (bit level) striping

Single copies of each strip are maintained. Error correcting code such as hamming code is calculated for the corresponding bits on each data disk.

The bits of code are stored in the corresponding bit positions on multiple parity disks.

The strips are very small so when a block is read all disks are accessed in parallel.



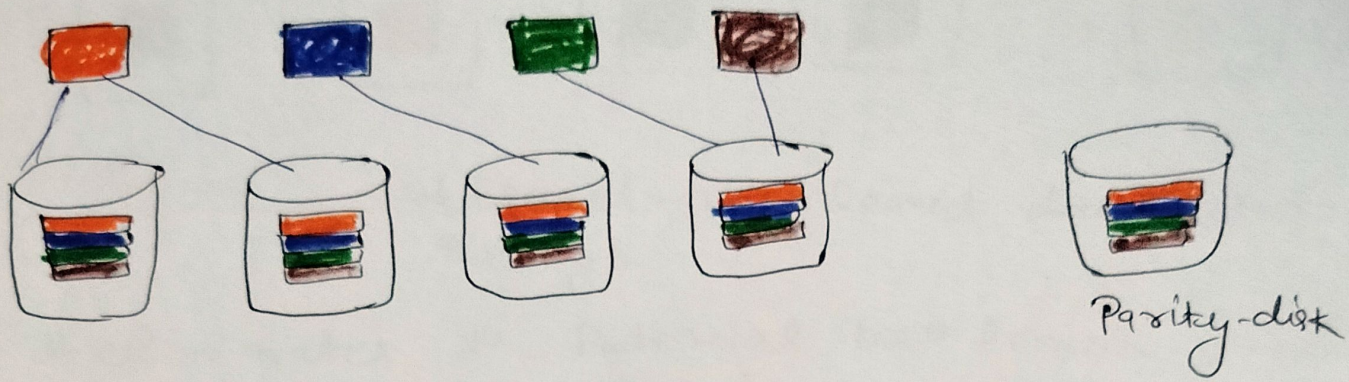
Advantages :- Good reliability with higher storage utilization than mirroring.

disadvantages :- Unnecessary cost - disk can already detect failure. Poor random performance.

RAID level-3 : (bit interleaved Parity)

In RAID level-3 single Parity bit is used instead of an error correcting code. It requires just one extra disk. The data stripping is used similar to other RAID levels.

if any disk in the array fails its data can be determined from the data on the remaining disks.



Advantages: Same Reliability with one disk failure as RAID-2 Since disk Controller can determine what disk failure.
Higher Storage utilization.

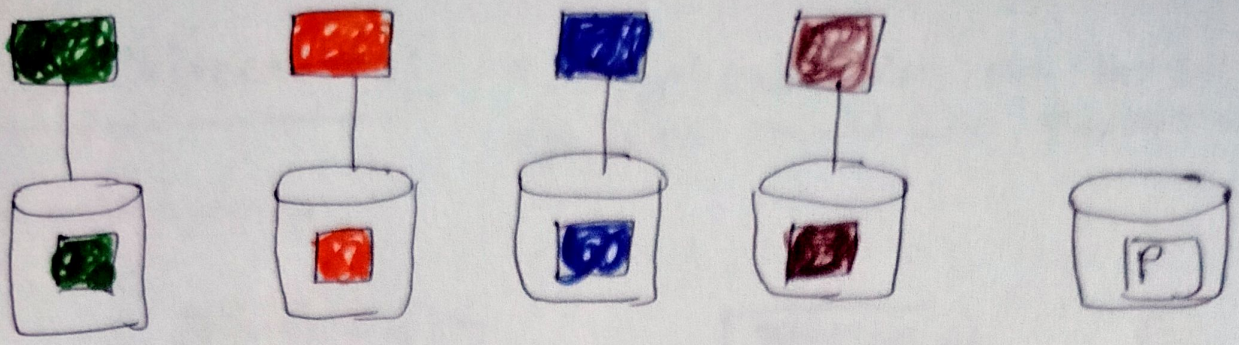
Disadvantages:- Poor random performance.

RAID Level-4: block-interleaved Parity byte level.

RAID level-4 is similar to RAID level-3 except strips are larger.
operation to read a block invalves only a single disk.

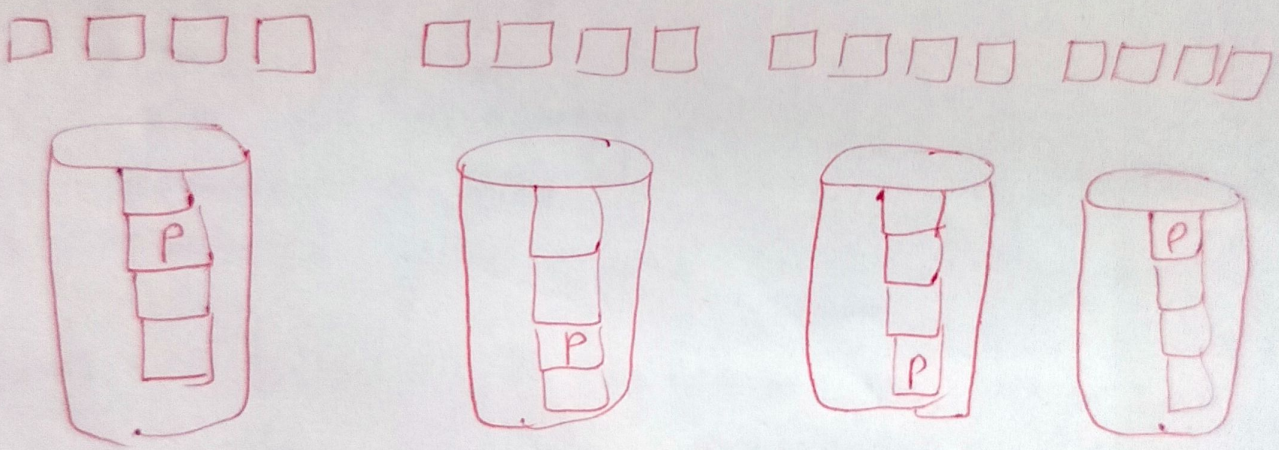
Parity bits are stored in the corresponding strip on the parity disk.

A bit by bit parity strip is calculated across corresponding data blocks on each data disk.

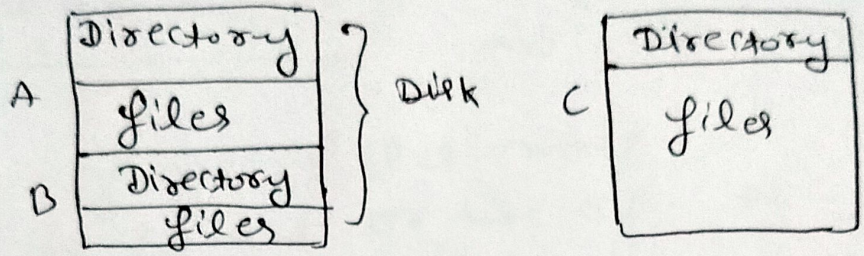


RAID-5' (Blocked - interleaved distributed Parity)

It eliminates the Potential bottleneck found in RAID-4. RAID-5 distributes the Parity strips across all disks.



FILE Directories!- A Physical disk can be broken up into multiple partitions or mini disks.



operations on directory!-

1) Search for a file
2) create New file
3) Delete a file
4) List a directory
5) Rename a file.
6) Traverse file system

Search!- Directory Structure is searched for finding Particular file in the directory files have symbolic names and similar names may indicate a relationship between files.

create a file!- when a new file is created an entry must be added to the directory.

(3) Delete a file!- when a file is deleted an entry must be removed from the directory.

(4) Rename a file!- when a file is deleted an entry must be

Name of the files must be changeable when the content or use of the file changes Renaming a file may also allow its position within the directory structure to be changed.

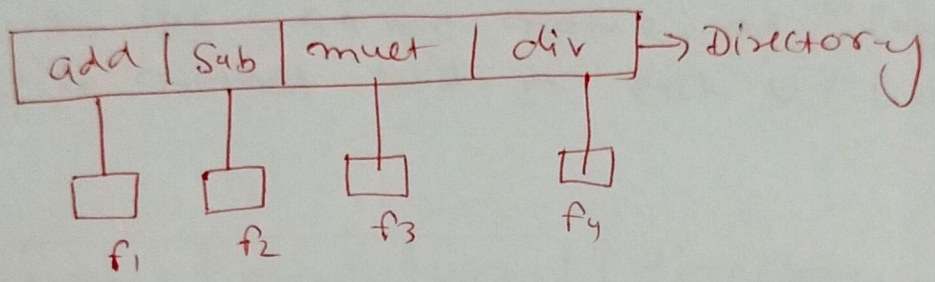
List directory!- All or portion of the directory may be requested. Request is made by a user and result in a listing of all files owned by that user plus some of the attributes of each file.

Different types of directory structure are given below.

(1) single level directory
(2) two level directory
(3) Tree structure directory
(4) Acyclic graph directory
(5) General graph directory.

(1) Single Level directory:

Single level directory structure is simple structure. All files are contained in the same directory easy to implement and maintain.



Advantages:

A single level directory structure is simple directory structure

(2) All files are contained in the same directory.

(3) Easy to implement.

Disadvantages:

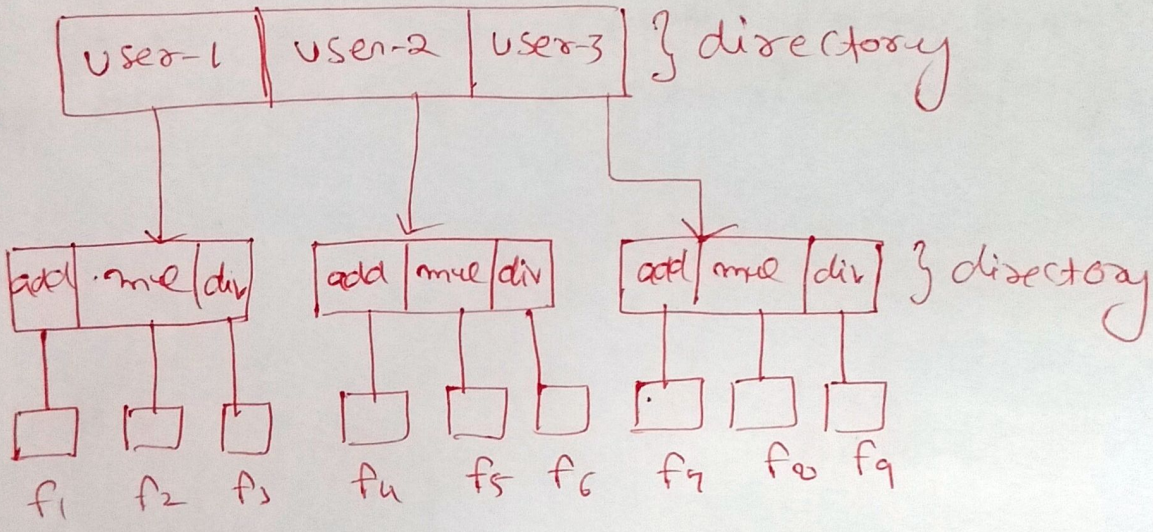
(1) Not suitable for a large number of files and more than one user.

(2) Because of single directory files require unique file name.

(3) It is difficult to remember the names of all the files as the number of files increases.

Two level directory:-

In two level directory each user has his own directory. It is called user file directory (UFD). Each user file directory has a similar structure. when a user refers to a particular file, only his own UFD is searched. Different users may have files with the same name, as long as all the file names with in each UFD are unique.



To create a file for a user the operating system searches only that users directory to ascertain whether another file of that name exists. to delete a file the operating system confines the search to the local UFD.

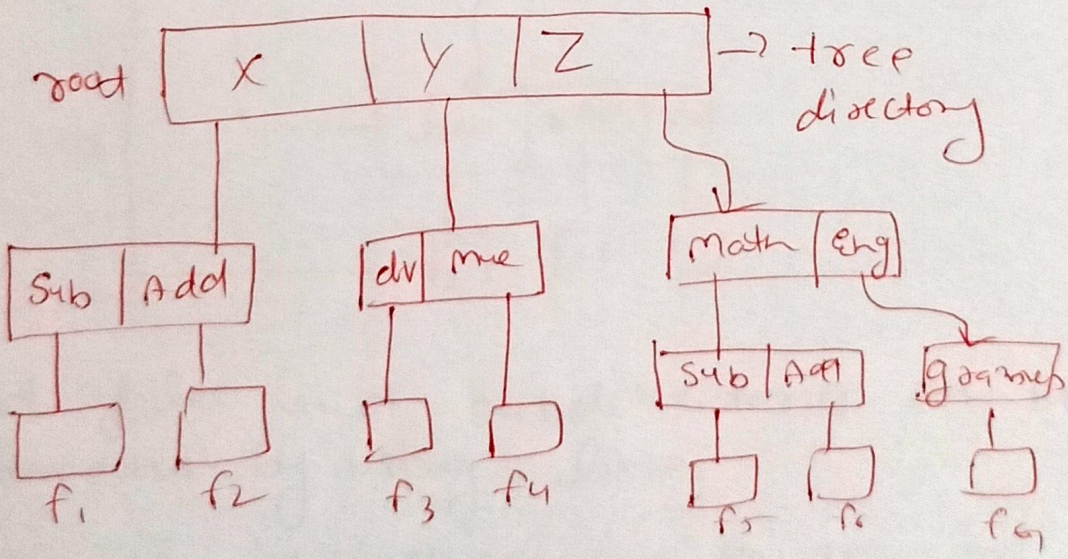
Operating System can not accidently delete another users file that has the same name.

Tree Structured Directories!

MS-DOS system is a tree structure directory. It allows users to create their own sub-directory and to organize their files accordingly. A subdirectory contains a set of files or subdirectories.

A directory is simple another file, but it is treated in a special way.

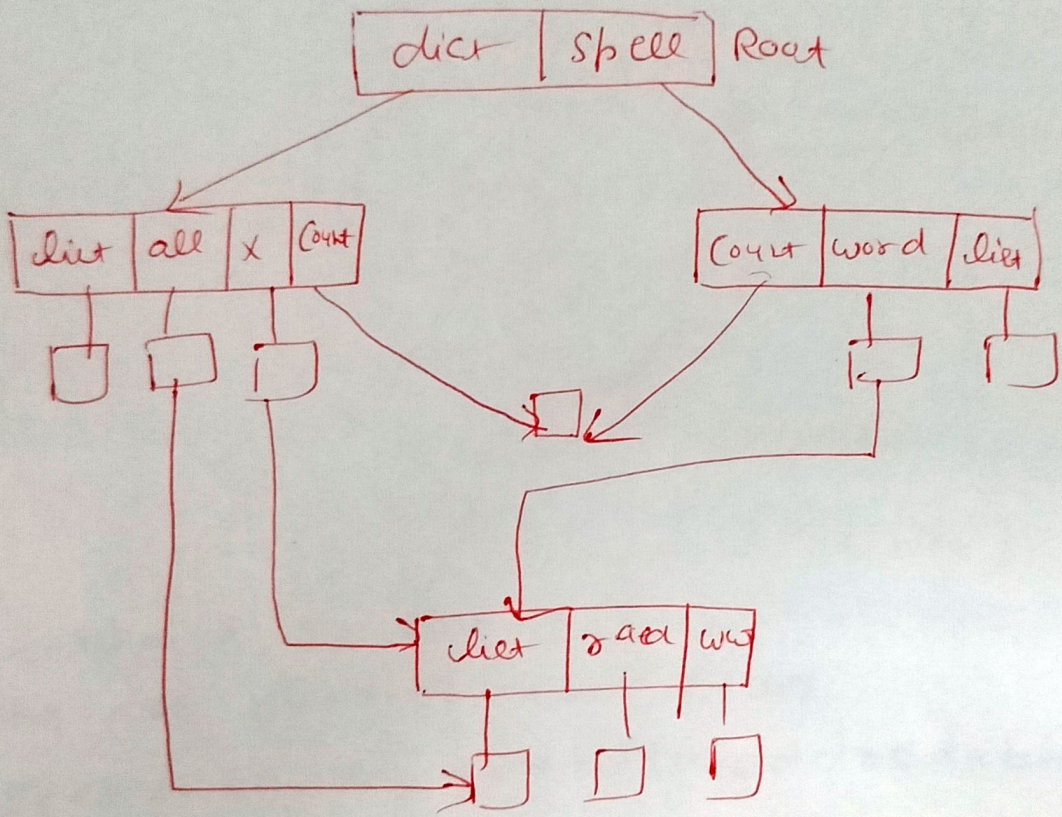
Path:- It is the path (route) from the root throught all the sub-directories to specified file.



Acyclic graph directories!

It allows directories to have shared subdirectories and files.

Same file or directory may be in two different directories.
Useful when the same files need to be accessed in more than one place in the directory structure.
files are shared by more than one user Progress.

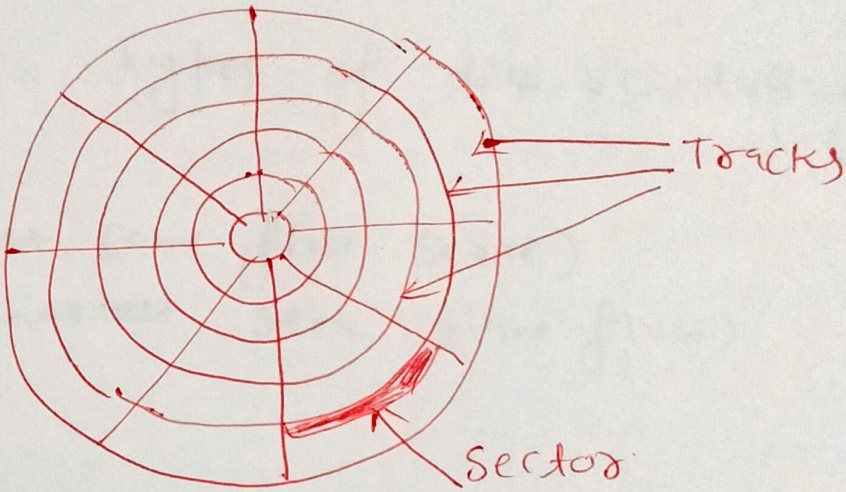


Shared files and subdirectories can be implemented by using links.
A link is effectively a pointer to another file or subdirectory.

Disk Storage:-

Magnetic disk: magnetic disks Provide bulk of data of secondary storage of modern computers.

A disk head reads and writes bits of data as they Pass under the head.



(Tracks and Sectors.)

Disk contains concentric track.
Tracks are divided into Sector
A Sector is the smallest addressable unit in a disk.

Transfer rate: Transfer rate is rate at which data flow between driven and computers.

Disk Scheduling Algorithm:-

A Process makes the Input/output requests to the operating System to access the disk Disk scheduling algorithm manages those requests and decides the order of the disk access given to the requests.

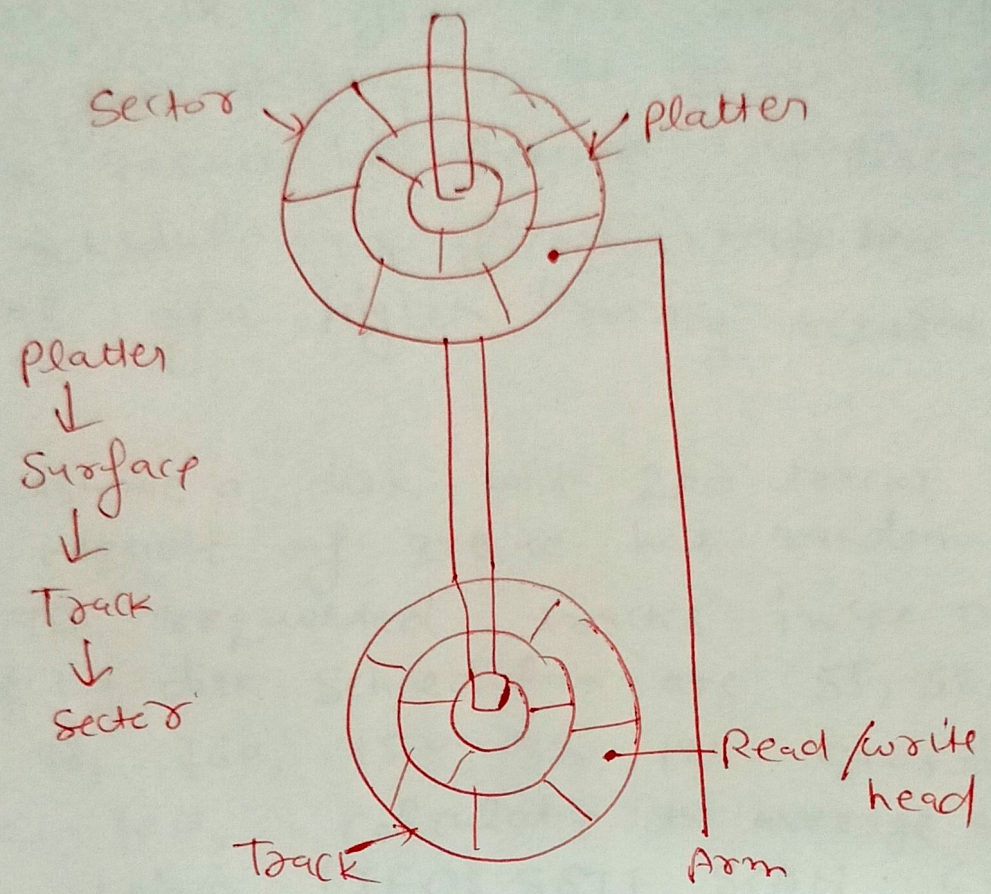
There are six types of disk scheduling algorithm.

(1) FCFS (first come first serve)
(2) SSTFC Shortest seek time first)
(3) SCAN
(4) C-SCAN
(5) LOOK
(6) C-LOOK

Term related to the Disk Scheduling

Seek time: It is the time taken by the disk arm to Locate the desired track.

Rotational Latency: The time taken by a desired sector of the disk to rotate itself to the position where it can access the Read/write head.



Transfer rate:- It is the time taken to transfer the data requested by the Process.

Disk access time:- Disk Access time is the sum of the seek time Rotational latency and transfer time.

FCFS (first-come-first serve) Scheduling

This is the simplest form of disk scheduling in which the first reque

to arrive is the first one serviced. FCFS has a fair Policy in the sense that once a request has arrived it place in the schedule is fixed irrespective of arrival of a Higher Priority request.

Ex-1

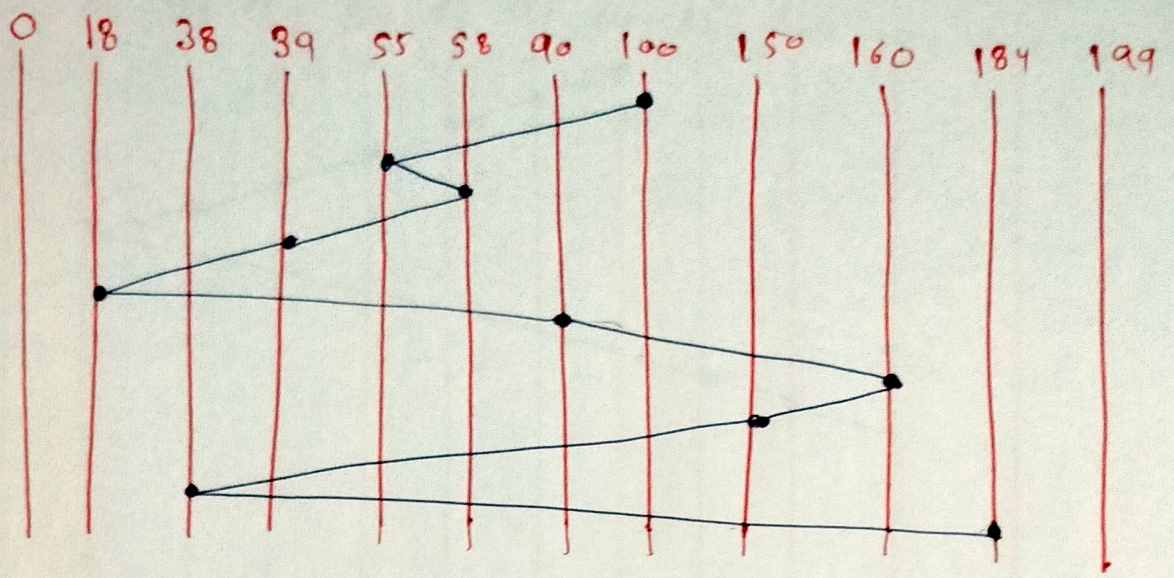
We assume a disk with 200 tracks and disk request of queue has random request int. The requested tracks in the order recived by disk Scheduler are 55, 58, 39, 18, 90, 160, 150, 38, 184 Starting at track 100, calculate the average seek length using FIFO, SSTF, SCAN, C-SCAN LOOK and C-LOOK give which disk Scheduling is best for the Scenario.

Request) (55, 58, 39, 18, 90, 160, 150, 38, 184)

queue(0, 55, 58, 39, 18, 90, 160, 150, 38, 184,
Starting at track = 100 (99)

Q4 (1) FCFS Scheduling

first we given sequence in acceding order.



average seek time = (100 − 55) + (58 − 55) + (58 − 39)
+ (39 − 18) + (90 − 18) + (160 − 90)
+ (160 − 150) + (150 − 38) + (184 − 38)

= 45 + 3 + 19 + 21 + 72 + 70 + 10 + 112
+ 146

= 498

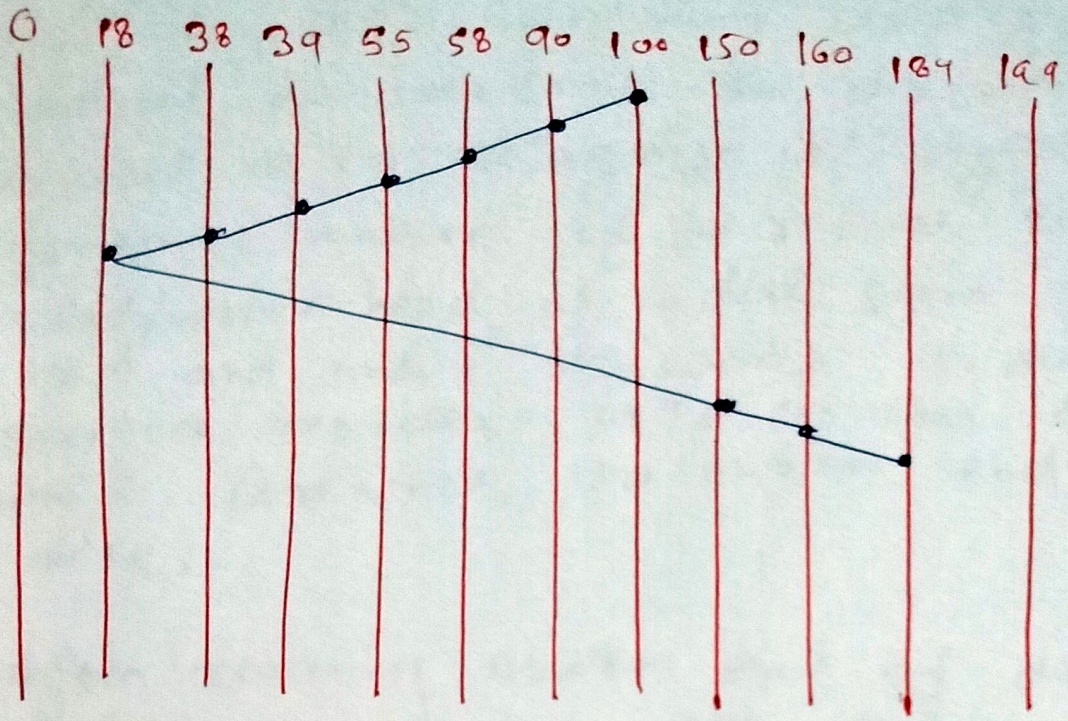
(2) SSTF (Shortest seek time first) Scheduling:

In Shortest seek time first SSTF scheduling Priority is given to those Processes which need the Shortest seek.

Request (55, 58, 39, 18, 90, 160, 150, 38, 184).

Queue (0, 55, 58, 39, 18, 90, 160, 150, 38, 184, 199)

Starting track = 100



Average seek time:

Seek time = (100 − 90) + (90 − 58) + (58 − 55) +
(55 − 39) + (39 − 38) + (38 − 18)
+ (150 − 18) + (160 − 150) + (184 − 160)

= 10 + 42 + 3 + 16 + 1 + 20
+ 132 + 10 + 24

= 55 + 37 + 132 + 34

= 92 + 166

= 258

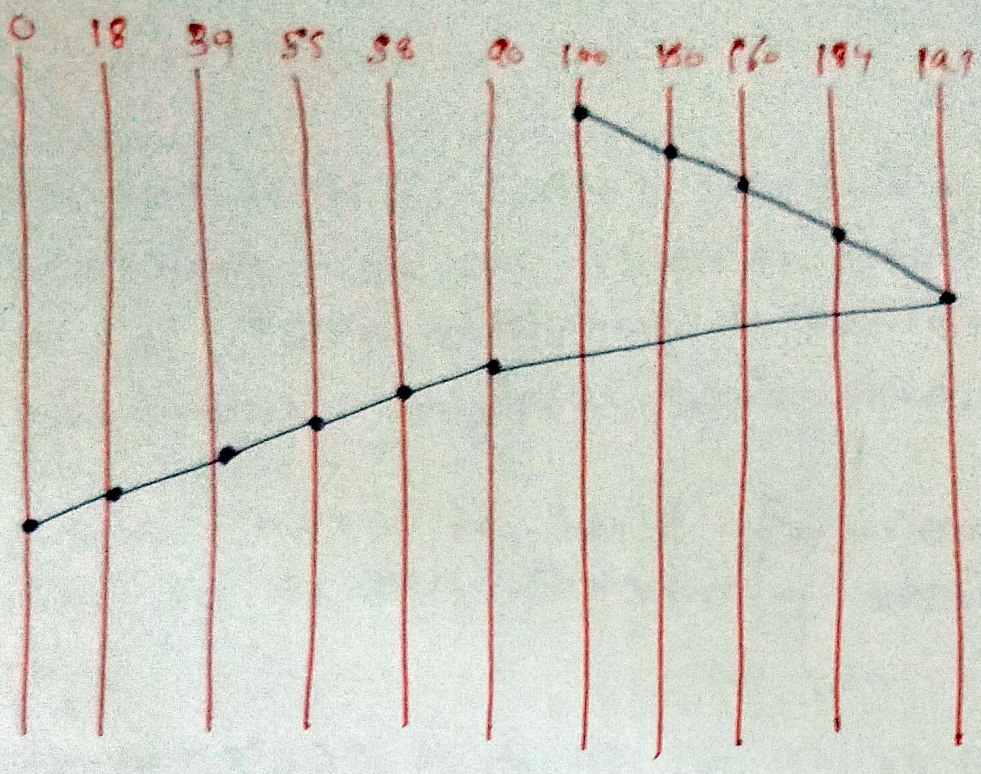
(3) SCAN: SCAN Scheduling strategy was developed to overcome the discrimination as well as to recognize the ~~dynamic~~ dynamic nature of the request queue. The read/write head of a disk start from one end and move towards the other end, services requests as it reaches each track until it reaches to another end of the disk.

After reaching another end of disk, disk head reverses its path-direction while continuing with services whichever comes on the way. This way disk head continuously oscillates from end-to end.

Request: (55, 58, 39, 18, 90, 160, 150, 38, 184)

Queue: (0, 55, 58, 39, 18, 90, 160, 150, 38, 184, 199)

Starting Point ⇒ 100



Average Seek time = (150 − 100) + (160 − 150) +

(184 − 160) + (199 − 184) +

(199 − 90) + (90 − 58) + (58 − 55) +

(55 − 39) + (39 − 18) + (18 − 0)

= 50 + 10 + 24 + 15 + 109 +

42 + 3 + 16 + 18 + 18

= 84 + 114 + 45 + 52

= 198 + 97

Seek time. = 295

C-SCAN (Scheduling Algo);-

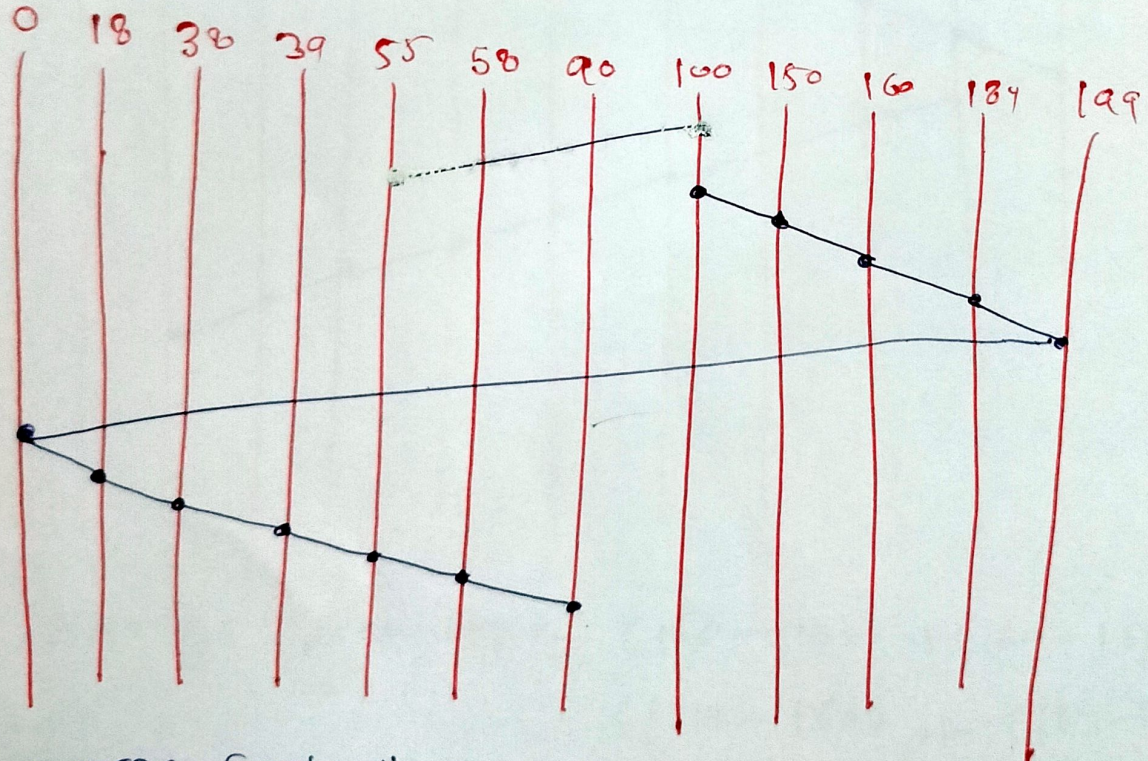
Circular SCAN (C-SCAN) scheduling restricts scanning to one direction only. It is a variant of SCAN designed to provide a more ~~information~~ uniform wait time.

C-SCAN moves the head from one end of the disk to the other servicing requests along the way.

Request (55, 58, 39, 18, 90, 160, 150, 38, 184)

Queue (0, 55, 58, 39, 18, 90, 160, 150, 38, 184, 199)

Starting point = 100



average seek time = (150 - 100) + (160 - 150) + (184 - 160) + (199 - 184) + (199 - 0) + (18 - 0) + (38 - 18) + (39 - 38) + (55 - 39) + (58 - 55) + (90 - 58)

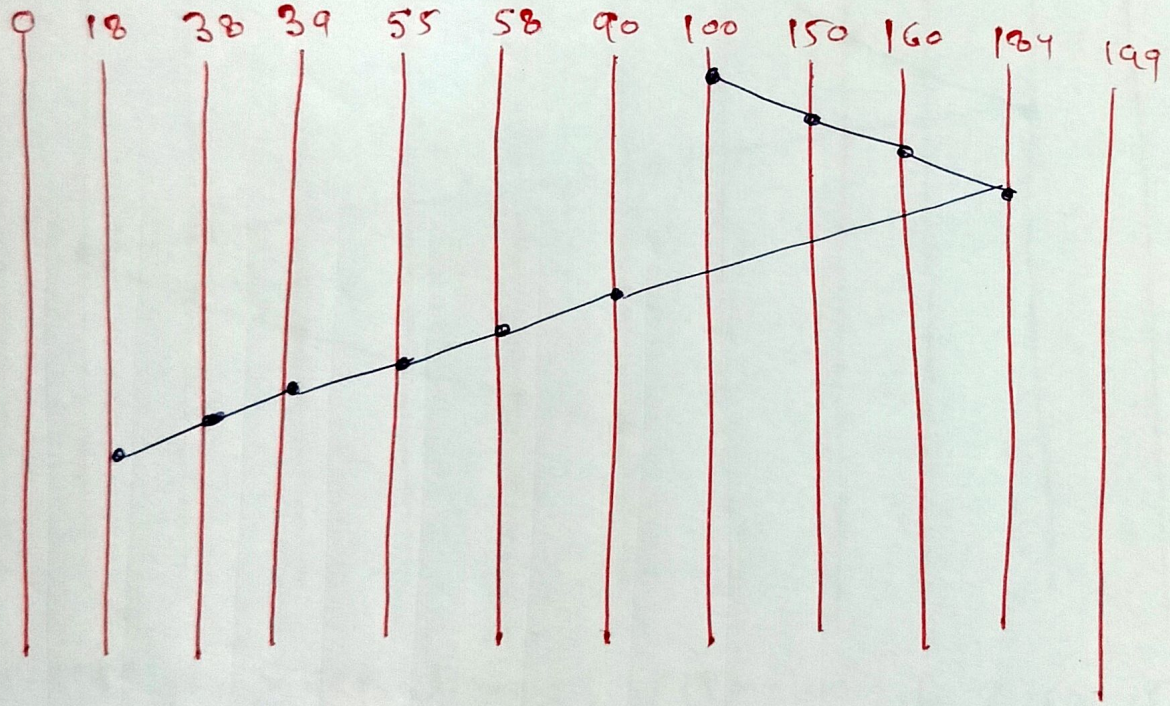
Look Scheduling Algorithm:

Start the head moving in one direction. Satisfy the request for the closest track in that direction when there is no more request in the direction, the head is traveling, reverse direction and repeat. This algorithm is similar to SCAN

Request (55, 58, 39, 18, 90, 160, 150, 38, 184,

quere (0, 55, 58, 39, 18, 90, 160, 150, 38, 184, 199)

Starting point = 100



Average Seek time = (150 − 100) + (160 − 150) + (184 − 160) + (184 − 90) + (90 − 58) + (58 − 57) + (55 − 39) + (39 − 38) + (38 − 18)

Seek time = 250

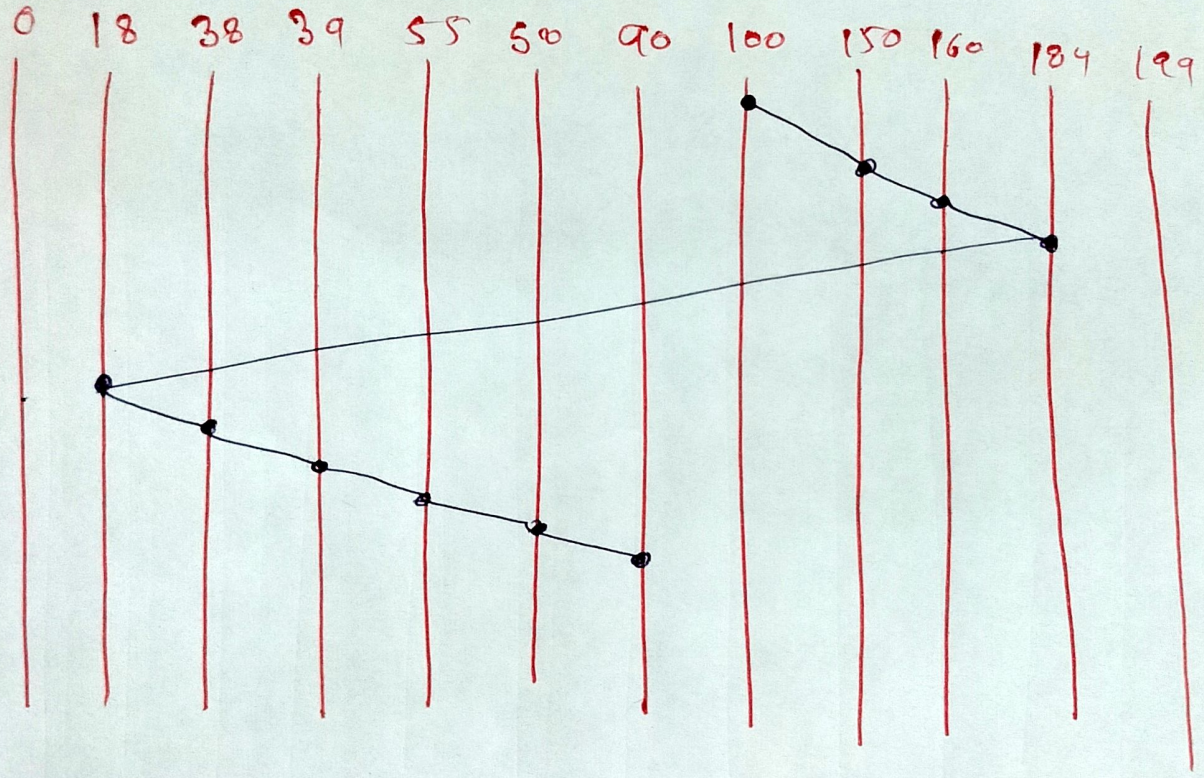
C-Look Scheduling :-

Version of C-SCAN, Arm only goes as far as the last request in each direction, then reverse direction immediately, without first going all the way to the end of the disk.

Request (55, 58, 39, 18, 90, 160, 150, 38, 184)

Queue (0, 55, 58, 39, 18, 90, 160, 150, 38, 184, 199)

starting point = 100



Average Seek time = (150-100) + (160-150) + (184-160) + (184-18) + (38-18) + (39-38) + (55-39) + (58-55) + (90-58)

= 322