UNIT-12

DISK MANAGEMENT

A computer system consists of several devices (such as mouse, keyboard, disk, monitor, CD-ROM) that deal with different I/O activities. Among all these I/O devices, disk is considered as an essential requirement for almost all the computers.

▪ Disk Structure:

A magnetic disk or hard disk is the most commonly used secondary storage medium. It offers high storage capacity and reliability. Whenever the data stored on the disk needs to be accessed by CPU, it is first moved to the main memory and then required operation is performed. Once the operation has been performed, the modified data must be copied back to the disk. The system (o.s) is responsible for transferring the data between the disk and the main memory as and when required. Data on the disk survives power failures and system crash. There is a chance that disk may sometimes fail itself and destroy the data, however, such failures occur rarely.

Data is represented as magnetized spots on a disk. A magnetized spot represents 1 and the absence of a magnetized spot represents 0. To read the data, the magnetized spots on the disk are converted into electrical impluses, which are then transferred to the processor. Writing data onto the disk is accomplished by converting the electrical impluses received from the processor into magnetized spots on the disk. The data in a magnetic disk can be erased and reused virtually infinitely.

A hard disk is a collection of platters, each disk platter has a flat circular shape, like a compactdisk(CD). Common platters diameter range from 1.8 to5.25 inches. The two surface of a platter are covered with a magnetic material. We Store information by recording it magnetically on the platters.

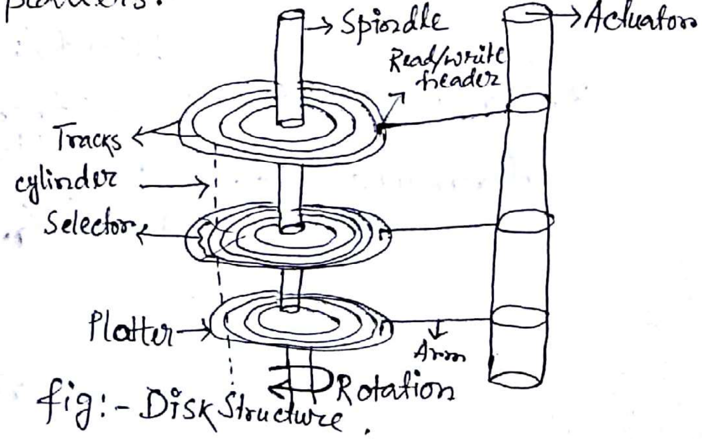


fig:- Disk Structure.

A read/write head located just above each surface of every platter. The space of platter is logically divided into circular 'Tracks'. The tracks are subdivided in to 'Sectors'. The set of tracks that are at one arm position forms a 'cylinder'. The heads are attached to a disk arm, which all the heads as a unit. The disk platters mounted on a 'spindle' together with the heads. Accessing data of one cylinder is much faster than accessing data that is distributed among different cylinders.

■ Hard Disk Performance Parameters – Terminologies:

1. Seek time : It is defined as the time required to move the disk arm to the required track.

It consists of two key components:—

(a) The initial startup time

(b) The time taken to traverse the track that have to be crossed ~~to~~ once the access arm is up to ~~seek~~ speed.

The traversal time is not a linear function of the number of tracks but includes a startup time and a settling time i.e., the time after positioning the head over the target track 'until track identification is confirmed.

The linear formulas for the seek time is:—

$$T_s = m * n + S$$

where T_s : Estimated seek time

n : Number of track traversed.

m : Constant that depends on the disk drive

S : Start up time.

2. Rotational Delay : It is defined as the time required to reach the desired sector by the read/write head.

3. Transfer Time : The transfer time is depends on the rotation speed of the disk. The formular is:–

$T = b/rN$. where T = transfer time

b = number of bytes to be transferred.

N = number of bytes on a track.

r = rotation speed in revolutions/second.

The total average access time is expressed as —

$$T_a = T_s + \frac{1}{2r} + \frac{b}{rN}.$$

◪ Hard Disk Scheduling Algorithms:

When a process wants to do disk I/O, an OS call is made. This may take sometime. The process is put in a blocked state and the I/O request is sent to the device driver. If the disk is idle, the operation is started else if the disk is busy, servicing another request then it is added to a queue of requests. To do this there are number of disk scheduling algorithms available.

- First-Come, First-Serve (FCFS) Algorithm:

The principle is "The disk controller processes the I/O requests in the order in which they arrive, thereby moving backwards and forwards across the surface of the disk to get to the next requested location each time".

Since no recording of request takes place so the head may move almost randomly across the surface of the disk. This algorithm minimized the response time.

Example

A disk queue has the following requests to read tracks—

87, 170, 40, 150, 36, 72, 66, 15.

If the disk head is initially at cylinder 60. Then it will first move from 60 to 87, then 87 to 170, 170 to 40, 40 to 150, 150 to 36, 36 to 72, 72 to 66, 66 to 15.

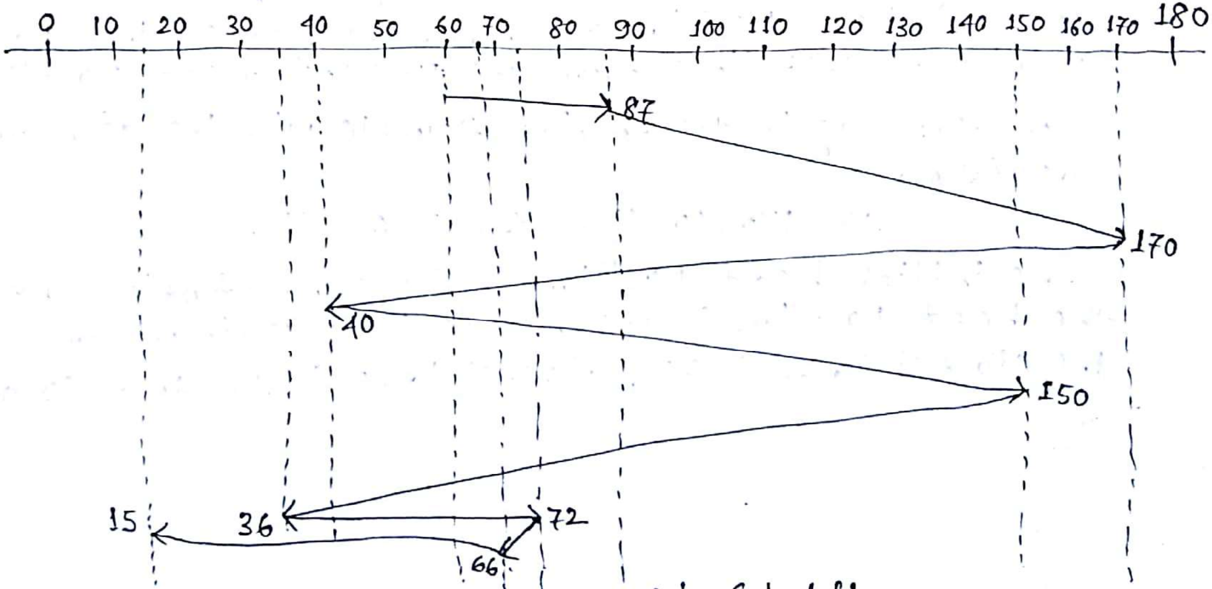


fig:- FCFS Disk Scheduling

∴ Total head movement is given by —
(60 to 87) + (87 to 170) + (170 to 40) + (40 to 150) + (150 to 36) + (36 to 72) + (72 to 66) + (66 to 15)
= (87 − 60) + (170 − 87) + (170 − 40) + (150 − 40) + (150 − 36) + (72 − 36) + (72 − 66) + (66 − 15).
= (27 + 83 + 130 + 110 + 114 + 36 + 6 + 51)
= 557 cylinders.

∴ Average head movement = 557/8 = 69.6 cylinders.

Advantages of FCFS :

1. It is a very simple algorithm to implement.
2. Improved response time as a request gets response in fair amount of time.

Disadvantages of FCFS:

1. It involves a lot of random head movements of disk rotations.
2. Throughput is not efficient.
3. It is used in small systems only where I/O efficiency is not very important.
4. FCFS is acceptable when the load on a disk is light. As the load grows, FCFS tends to saturate the device and the response time becomes longer.

- Shortest Seek time First (SSTF):

This algorithm works on this principle – "When a disk operation finishes, choose the request that is closest to the current head position or choose the request that has minimum seek time from the current head position".

Consider again the previous example with the disk queue as follows —

87, 170, 40, 150, 36, 72, 66, 15.

The initial head position in 60. Now, ~~closest~~ closest to the head position is the request at cylinder 66. Then the closest to 66 is 72, closest to 72 is 87 and so on.

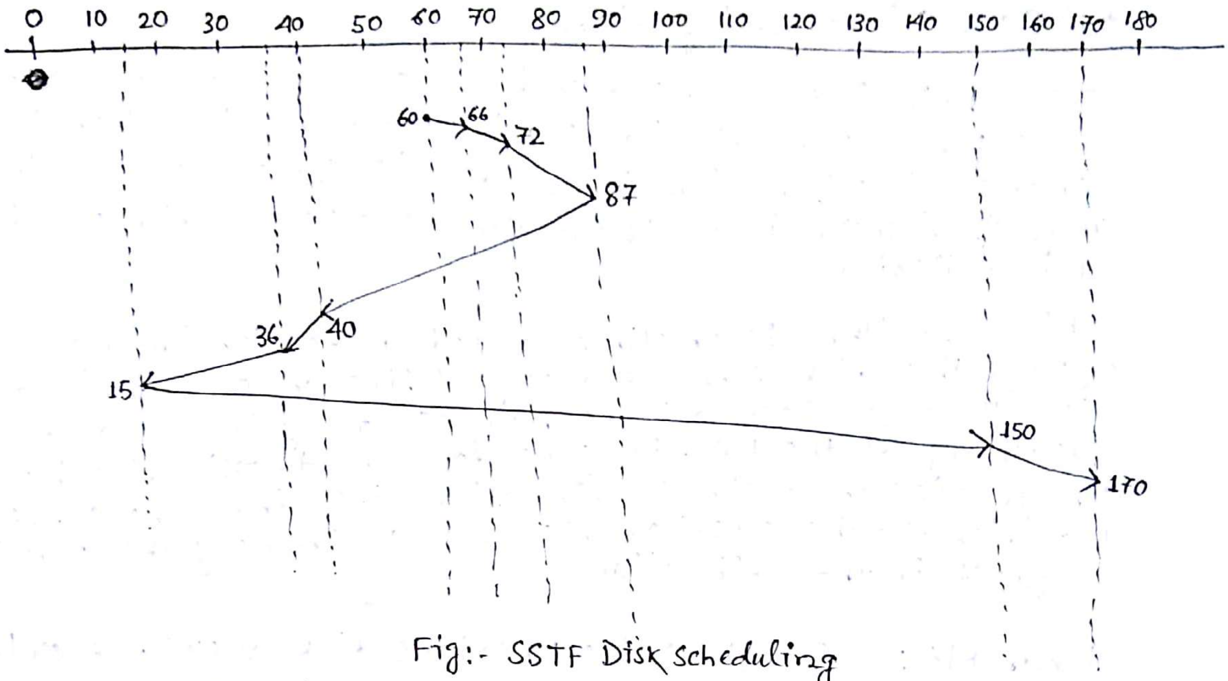


Fig:- SSTF Disk Scheduling

∴ Total head movements in SSTF are —

$= (60 \text{ to } 66) + (66 \text{ to } 72) + (72 \text{ to } 87) + (87 \text{ to } 40) + (40 \text{ to } 36) + (36 \text{ to } 15)$
$+ (15 \text{ to } 150) + (150 \text{ to } 170)$

$= (66-60) + (72-66) + (87-72) + (87-40) + (40-36) + (36-15)$
$+ (150-15) + (170-150)$

$= (6+6+15+47+4+21+135+20) = 254$ cylinders.

∴ Average head movements are $= 254/8 =$ ~~30.5~~ cylinders.
31.8

Advantages of SSTF:

1. It minimized latency.
2. Better throughput than FCFS method.

Disadvantages of SSTF:

1. Starvation may occur here. As we know that requests arrive at random in a real system. Some process may have to wait for a long time until its requests are satisfied, if new requests with shorter seek time keep arriving. This may cause starvation of some requests.
2. SSTF services requests for those tracks which are highly localized. So, the innermost and outermost tracks received poor service as compared to the midrange tracks.

• Scan/Elevator Algorithm:

The principle : " The disk head constantly moves from the most inner cylinder to the outer cylinder and then it changes its direction back towards the center. As the head moves, if there is a request for the current disk position then it is satisfied".

This algorithm is sometimes called as the elevator algorithm, because the disk arm behaves just like an elevator of a building, firstly it services all the request of going up and then reaching at the top, it goes downward. The disk head progresses in a single direction, i.e., from the center of the disk to the edge or vice versa, serving the closet request in that direction.

Example :- Consider the previous disk queue. (except40)

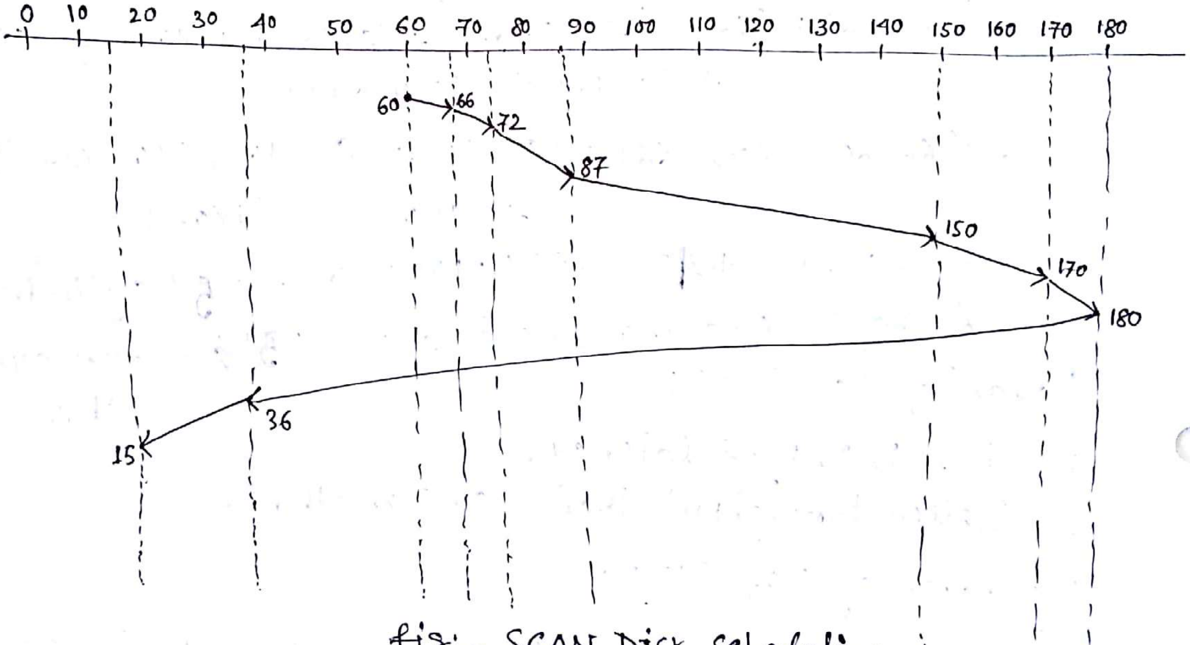


fig:- SCAN Disk Scheduling

∴ Total head movement $= |(66-60)| + |(72-66)| + |(87-72)| + |(150-87)| + |(170-150)| + |(180-170)| + |(180-36)| + |(36-15)|$

$= 6+6+15+63+20+10+144+21$

$= 285$ cylinders.

∴ Average movements $= 285/8 = 35.6$ cylinders.

Advantages of SCAN:

1. The througput is betterthan FCFS..
2. It has been the basis of most disk scheduling strategies.
3. It eliminates the discrimination inherent in SSTF schemes.

Disadvantages of SCAN:

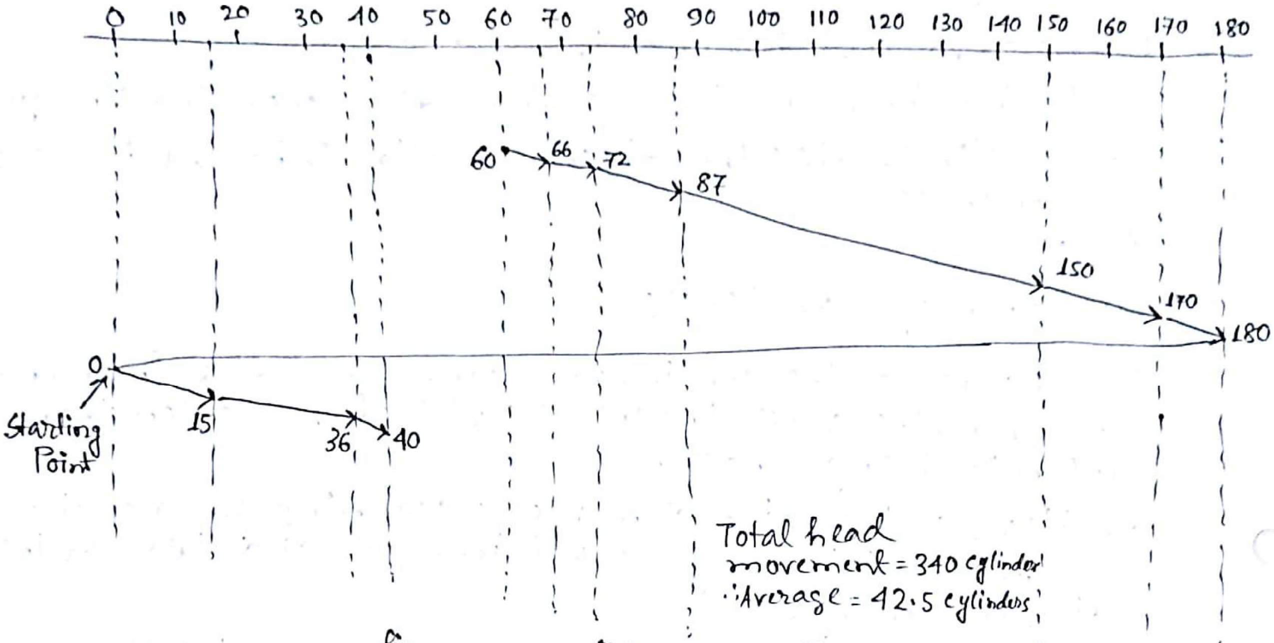
1. Because of the continuous scanning of disk from end to end, the outer tracks are visited less often than the mid range track.
2. Also, as the disk arm keeps scanning between two extremes, this may result in wear and tear of the disk assembly.
3. Certain requests arriving ahead of the arm position would get immediate service but some other requests that arrive behind the arm position will have to wait for the arm to return back. So, this algorithm is not fair.

- ## C-SCAN / One way Elevator Algorithm:

C-SCAN stands for circular SCAN and it is so called because this algorithm treats the cylinder as a circular list.

Principle: "The head sweeps from the innermost cylinder to the outmost cylinder satisfying the waiting requests in order of their locations. When it reaches the ~~outm~~ outermost cylinder, it sweeps back to the innermost cylinder without satisfying any requests and then starts again".

Example:- consider the previous disk queue again.



Total head movement = 340 cylinder
∴ Average = 42.5 cylinders

fig: C-SCAN Disk Scheduling

Advantages of C-SCAN:

1. It is more fair as compared to SCAN.
2. It provides more uniform waiting time.

Disadvantages of C-SCAN:

1. The time taken for the back swing has been ignored.
2. The average head movements in this algorithm is more as compared to SCAN algorithm.
3. This method increases the total seek time because of the long seek from the edge back to the hub.

- Look/Seek Algorithm:

Principle — "The drive sweeps across the surface of the disk in alternating directions, satisfying requests. But now, the drive makes use of the information it has about the locations requested by the waiting requests."

For instance, a sweep out, towards the outer edge of the disk will be reversed when there are no waiting requests for locations beyond the current cylinder. This improves both, throughput and the response time.