AOS Assignment No. 5

❖ 2 Marks Questions

Q.1.) What is goal of Disk Scheduling in OS?

Ans :- The goal of **disk scheduling** in an operating system is to:

- **Reduce the total seek time** (time taken to move the disk arm to the track where data is located).
- **Improve disk throughput** by servicing more requests in less time.
- **Ensure fair and efficient access** to the disk for all processes.
- **Optimize overall system performance** by minimizing I/O wait time.

Q.2.) Define seek time in Disk scheduling.

Ans :- **Seek time** in disk scheduling is the time it takes for the read/write head of the disk to move to the track where the requested data is stored.

It is one of the main components of disk access time, and reducing seek time is a key objective in disk scheduling to improve performance.

Q.3.) What is Transfer Time & Disk Access Time?

Ans :- Transfer time is the time required to **actually transfer data** between the disk and memory once the read/write head is in position.
It depends on the data transfer rate and the amount of data being transferred.

Disk Access Time

Disk access time is the **total time** it takes to access data on a disk. It includes:

1. **Seek Time** – Time to move the read/write head to the correct track.

2. **Rotational Latency** – Time waiting for the desired sector to rotate under the read/write head.
3. **Transfer Time** – Time to transfer the data once the head is in position.

Formula:
Disk Access Time = Seek Time + Rotational Latency + Transfer Time

Q.4.) What are the disadvantages of FCFS Disk Scheduling Algorithm?

Ans :- **Disadvantages of FCFS (First-Come, First-Served) Disk Scheduling Algorithm:**

1. **High Average Seek Time:**
   Requests may be scattered all over the disk, causing long head movements and increased seek time.
2. **Poor Performance:**
   It doesn't optimize head movement, leading to inefficient disk usage.
3. **No Prioritization:**
   All requests are treated equally, regardless of their location or urgency.
4. **Long Waiting Time for Some Requests:**
   A request far from the current head position may wait a long time if earlier requests are on opposite ends.
5. **Not Suitable for Heavy Load:**
   Under heavy disk traffic, performance can degrade significantly.

Q.5.) What do you mean by disk response time?

Ans :- **Disk Response Time** refers to the **total time taken by the disk to complete an I/O request**—from the moment the request is made until the data is transferred.

It includes:

1. **Seek Time** – Time to move the read/write head to the desired track.

2. **Rotational Latency** – Time for the desired sector to rotate under the head.
3. **Transfer Time** – Time to transfer data to or from the disk.

In short:

Disk Response Time = Seek Time + Rotational Latency + Transfer Time

It is a key factor in measuring the **performance of disk scheduling algorithms**.

❖ 10 Marks Questions

Q.1.) Explain SSTF Scheduling Algorithm with an example?

Ans :- **SSTF (Shortest Seek Time First) Scheduling Algorithm**

Definition:
SSTF selects the disk I/O request that is **closest to the current head position**, minimizing seek time for each step.

How it works:

- The disk head moves to the request that needs the **least movement** from its current position.
- It repeats this for the next closest request.

Example:

Given:

- Disk head starts at **position 50**
- Pending requests: **82, 170, 43, 140, 24, 16, 190**

Steps:

1. **Current head = 50**
   Closest request = **43** (|50 - 43| = 7)
   Move to 43
2. **Current head = 43**
   Closest request = **24** (|43 - 24| = 19)
   Move to 24
3. **Current head = 24**
   Closest request = **16** (|24 - 16| = 8)
   Move to 16
4. **Current head = 16**
   Closest request = **82** (|16 - 82| = 66)
   Move to 82
5. **Current head = 82**
   Closest request = **140** (|82 - 140| = 58)
   Move to 140
6. **Current head = 140**
   Closest request = **170**
   Move to 170
7. **Current head = 170**
   Move to **190**

Advantages:

- Better performance than FCFS.
- Reduces average seek time.

Disadvantages:

- Can lead to **starvation** of requests that are far from the current head position.

Q.2.) Explain the difference between SCAN & C-SCAN scheduling.

Ans :-

Difference between SCAN and C-SCAN Disk Scheduling Algorithms

Feature	SCAN (Elevator Algorithm)	C-SCAN (Circular SCAN)
Movement	Head moves in one direction servicing all requests, then **reverses** direction.	Head moves in one direction servicing requests, then **jumps to the start** and continues in same direction.
Direction Change	Yes, direction changes at the ends.	No, head returns to start without servicing during return.
Fairness	Less fair for requests at the ends.	More uniform wait time for all requests.
Performance	Slightly better than FCFS and SSTF.	More consistent performance compared to SCAN.
Example Analogy	Like an **elevator** that moves up and down.	Like a **circular elevator** that only moves up, then jumps to ground floor to start again.

Q.3.) With suitable example define the Look Algorithm?

Ans :- **LOOK Disk Scheduling Algorithm**

Definition:
The **LOOK algorithm** is a version of the SCAN algorithm. Instead of going all the way to the end of the disk, the disk arm **"looks ahead"** and only goes as far as the **last request in the current direction**, then reverses.

How it works:

- Head moves in one direction and services all requests.
- It **reverses direction** only when there are **no more requests** in that direction.

Example:

Given:

- **Initial head position = 53**
- **Requests = 98, 183, 37, 122, 14, 124, 65, 67**
- **Direction = moving toward higher cylinders**

Steps:

1. Arrange the requests: **14, 37, 53, 65, 67, 98, 122, 124, 183**
2. Start at **53**, moving right:
    - Service: **65 → 67 → 98 → 122 → 124 → 183**
3. Now no more requests to the right, so **reverse direction**:
    - Service: **37 → 14**

Head movement order:

53 → 65 → 67 → 98 → 122 → 124 → 183 → 37 → 14

Advantages:

- Reduces unnecessary movement to the end of the disk.
- More efficient than SCAN in terms of total seek time.

Disadvantages:

- May still cause longer wait times for requests far from the head in the opposite direction.

Q.4.) Suppose the following disk request sequence (track numbers) for a disk with 100 tracks is given: 45, 20, 90, 10, 50, 60, 80 and 70. Assume that the initial position of the R/W head is on track 50. Find the number of head movements in cylinders using SCAN Scheduling.

Ans :- Let's solve the problem step by step using the **SCAN Disk Scheduling Algorithm**.

Given:

- **Track requests:** 45, 20, 90, 10, 50, 60, 80, 70
- **Initial head position:** 50
- **Total tracks:** 0 to 99
- **Assume direction:** Moving **toward higher-numbered tracks** (right)

Step 1: Sort the requests

Sorted list: **10, 20, 45, 50, 60, 70, 80, 90**

Split based on head position (50):

- **Left of 50:** 10, 20, 45
- **At/Right of 50:** 50, 60, 70, 80, 90

Step 2: SCAN Scheduling (Right, then Left)

Head movement order:

1. Start at **50**
2. Go right: **60 → 70 → 80 → 90**
3. Reverse and go left: **45 → 20 → 10**

Step 3: Calculate head movements

- 50 → 60 = 10

- 60 → 70 = 10
- 70 → 80 = 10
- 80 → 90 = 10
- 90 → 45 = 45
- 45 → 20 = 25
- 20 → 10 = 10

Total Head Movements:

10 + 10 + 10 + 10 + 45 + 25 + 10 = 120 cylinders

Q.5.) Consider the following disk request sequence for a disk with 100 tracks 45, 21, 67, 90, 4, 50, 89,52, 61, 87, 25 Head pointer starting at 50 and moving in left direction. Find the number of head movements in cylinders using FCFS scheduling.

Ans :- Let's solve the problem using the **FCFS (First-Come, First-Served)** disk scheduling algorithm.

Given:

- **Track requests (in order):**
  45, 21, 67, 90, 4, 50, 89, 52, 61, 87, 25
- **Initial head position:** 50
- **Disk size:** 100 tracks (0 to 99)
- **Head direction:** Given as "left" but **FCFS ignores direction** — it services requests **in the given order**.

Step 1: Head movement order (as per FCFS):

Start at 50, then go to:
45 → 21 → 67 → 90 → 4 → 50 → 89 → 52 → 61 → 87 → 25

Step 2: Calculate head movements

From	To	Movement
50	45	5
45	21	24
21	67	46
67	90	23
90	4	86
4	50	46
50	89	39
89	52	37
52	61	9
61	87	26
87	25	62

Total Head Movements:

5 + 24 + 46 + 23 + 86 + 46 + 39 + 37 + 9 + 26 + 62 = 403 cylinders