Assignment 8: Write a program to simulate disk scheduling algorithms
FCFS
SCAN
SSTF

Disk Scheduling Algorithms

What is Disk Scheduling Algorithm?

A Process makes the I/O 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.

Why is the Disk Scheduling Algorithm needed?

Disk Scheduling Algorithms are needed because a process can make multiple I/O requests and multiple processes run at the same time. The requests made by a process may be located at different sectors on different tracks. Due to this, the seek time may increase more. These algorithms help in minimizing the seek time by ordering the requests made by the processes.

Important Terms related to Disk Scheduling Algorithms

1. **Seek Time:** As we know, the data may be stored on various blocks of disk. To access these data according to the request, the disk arm moves and finds the required block. The time taken by the arm in doing this search is known as "Seek Time"**.**
2. **Rotational Latency:** The required data block needs to move at a particular position from where the read/write head can fetch the data. So, the time taken in this movement is known as "Rotational Latency". This rotational time should be as less as possible so, the algorithm that will take less time to rotate will be considered a better algorithm.
3. **Transfer Time:** When a request is made from the user side, it takes some time to fetch these data and provide them as output. This time taken is known as "Transfer Time".
4. **Disk Access Time**: It is defined as the total time taken by all the above processes. Disk access time = (seek time + rotational latency time + transfer time)
5. **Disk Response Time:** The disk processes one request at a single time. So, the other requests wait in a queue to finish the ongoing process of request. The average of this waiting time is called "Disk Response Time".

6. **Starvation:** Starvation is defined as the situation in which a low-priority job keeps waiting for a long time to be executed. The system keeps sending high-priority jobs to the disk scheduler to execute first.

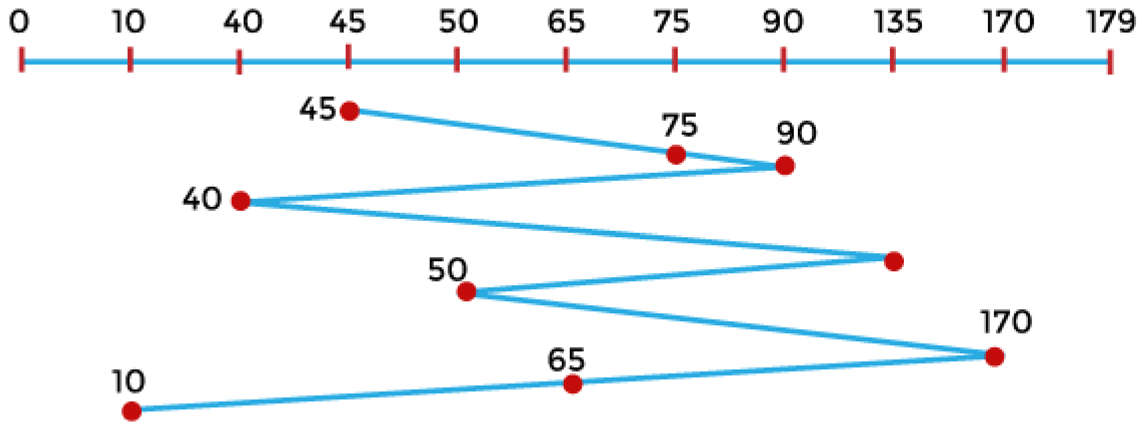
First Come First Serve (FCFS)

- FCFS stands for First-Come-First-Serve. It is a very easy algorithm among the all-disk scheduling algorithms.
- It is an OS disk scheduling algorithm that runs the queued requests and processes in the way that they arrive in the disk queue. It is a very easy and simple CPU scheduling algorithm.
- In this scheduling algorithm, the process which requests the processor first receives the processor allocation first. It is managed with a FIFO queue.

Example:

Let's take a disk with 180 tracks (0-179) and the disk queue having input/output requests in the following order: 75, 90, 40, 135, 50, 170, 65, 10. The initial head position of the Read/Write head is 45. Find the total number of track movements of the Read/Write head using the FCFS algorithm.

Solution:



FCFS Disk Scheduling Algorithm

Total head movements,
The initial head point is 45,
= (75-45) + (90-75) + (90-40) + (135-40) + (135-50) + (170-50) + (170-65) + (65-10)
= 30 + 15 + 50 + 95 + 85 + 120 + 105 + 55
= 555

Example 2 :

(NOTE: Please follow this output format for all the programs)

```
Enter the max range of disk
200
Enter the size of queue request
7
Enter the queue of disk positions to be read
82,170,43,140,24,16,190
Enter the initial head position 50
Total seek time is 642
Average seek time is 91.71
```

Advantages

- It is a very easy type of disk scheduling algorithm.
- It is easy to program.
- It provides a first come first served process.
- In FCFS, each process eventually has a chance to execute, therefore there is no starvation.

Disadvantages

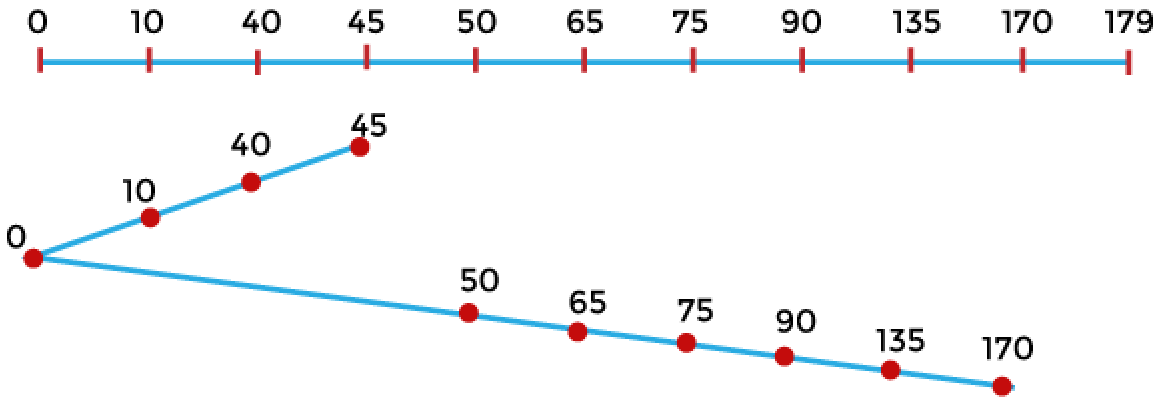
- It is not very efficient because of its simplicity.
- Its average waiting time is high.
- It is a Non-Preemptive CPU Scheduling Algorithm, which implies that once a process has been assigned to a CPU, it would never release the CPU until the process has completed executing.

SCAN

- It is also known as the Elevator algorithm.
- In this algorithm, the head may move in both directions, i.e., the disk arm begins to move from one end of the disk to the other end and servicing all requests until it reaches the other end of the disk.
- After reaching the other end, the head position direction is changed and further continues servicing the requests till the end of the disk.

Example:

Let's take a disk with 180 tracks (0-179) and the disk queue having input/output requests in the following order: 75, 90, 40, 135, 50, 170, 65, 10. The initial head position of the Read/Write head is 45 and will move on the left-hand side. Find the total number of track movements of the Read/Write head using the SCAN algorithm.



SCAN Disk Scheduling Algorithm

Solution:
Initial head point is 45,
= (45-40) + (40-10) + (10-0) + (50-0) + (65-50) + (75-65) + (90-75) + (135-90) + (170-135)
= 5 + 30 +10 +50 +15 + 10 +15 + 45 + 35
= 215

Advantages

- It is easy to use and understand.
- In the SCAN disk scheduling algorithm, low variance happens in the waiting time and response time.
- Starvation is avoided.

Disadvantage

- If no requests remain to be serviced, the head moves till the end of the disk.

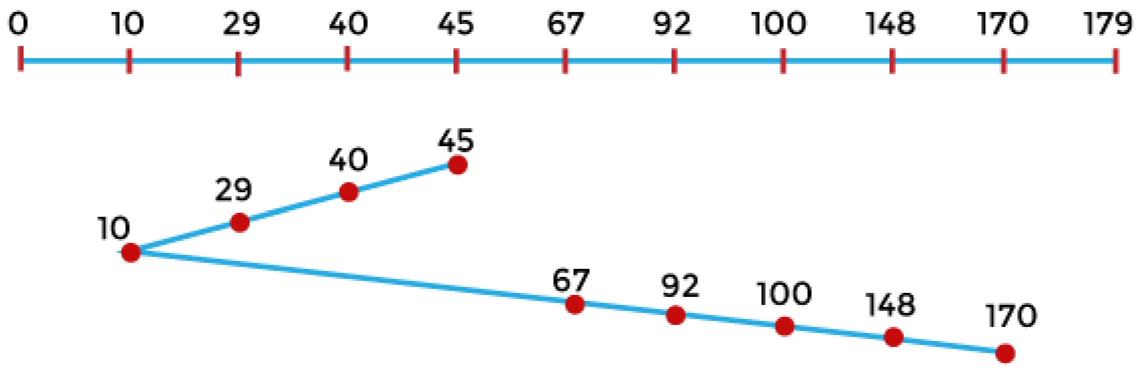
Shortest Seek Time First (SSTF):

SSTF stands for Shortest Seek Time First, and it serves the request that is closest to the current position of the head. The direction of the head pointer is quite important in this algorithm. If a tie happens between requests, the head will serve the request in its current direction. In comparison to the FCFS, the SSTF algorithm is very efficient in terms of the total seek time.

Example:
Let's take an example to understand the SSTF Disk Scheduling Algorithm. Let's take a disk with 180 tracks (0-179) and the disk queue having input/output requests in the

following order: 92, 100, 40, 148, 67, 170, 29, 10. The initial head position of the Read/Write head is 45 and will move in the left-hand side direction. Find the total number of track movements of the Read/Write head using the SSTF algorithm.

Solution:



SSTF Disk Scheduling Algorithm

Total head movements,
Initial head point is 45,
= (45-40) + (40-29) + (29-10) + (67-10) + (92-67) + (100-92) + (148-100) + (170-148)
= 5 + 11 + 19 + 57 + 25 + 8 + 48 + 22
= 195

Advantages

- It improves and increases the throughput.
- SSTF's total seek time is lower than the FCFS.
- It has less response time and average waiting time.

Disadvantages

- Starvation can happen for requests far from the head.
- In the SSTF disk scheduling algorithm, the high variance is available in response time and waiting time.
- The algorithm is slowed by frequent changes in the head's direction.