Disk Scheduling

As we know, a process needs two type of time, CPU time and IO time. For I/O, it requests the Operating system to access the disk.

However, the operating system must be fare enough to satisfy each request and at the same time, operating system must maintain the efficiency and speed of process execution.

The technique that operating system uses to determine the request which is to be satisfied next is called disk scheduling.

**Important terms related to disk scheduling:**

### **Seek Time**

Seek time is the time taken in locating the disk arm to a specified track where the read/write request will be satisfied.

### **Rotational Latency**

It is the time taken by the desired sector to rotate itself to the position from where it can access the R/W heads.

### **Transfer Time**

It is the time taken to transfer the data.

### **Disk Access Time**

Disk access time is given as,

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

### **Disk Response Time**

It is the average of time spent by each request waiting for the IO operation.

### **Purpose of Disk Scheduling**

The main purpose of disk scheduling algorithm is to select a disk request from the queue of IO requests and decide the schedule when this request will be processed.

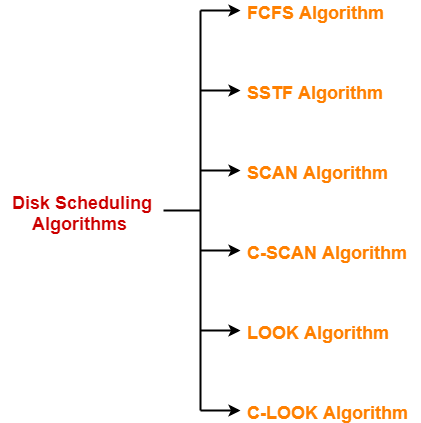
### **Goal of Disk Scheduling Algorithm**

* Fairness
* High throughout
* Minimal traveling head time

## **Disk Scheduling Algorithms-**

* The algorithms used for disk scheduling are called as **disk scheduling algorithms**.
* The purpose of disk scheduling algorithms is to reduce the total seek time.

Various disk scheduling algorithms are-



1. FCFS Algorithm
2. SSTF Algorithm
3. SCAN Algorithm
4. C-SCAN Algorithm
5. LOOK Algorithm
6. C-LOOK Algorithm

## **FCFS Disk Scheduling Algorithm-**

* As the name suggests, this algorithm entertains requests in the order they arrive in the disk queue.
* It is the simplest disk scheduling algorithm.

## **Advantages-**

* It is simple, easy to understand and implement.
* It does not cause starvation to any request.

## **Disadvantages-**

* It results in increased total seek time.
* It is inefficient.

## **SSTF Disk Scheduling Algorithm-**

* SSTF stands for **Shortest Seek Time First**.
* This algorithm services that request next which requires least number of head movements from its current position regardless of the direction.
* It breaks the tie in the direction of head movement.

## **Advantages-**

* It reduces the total seek time as compared to **FCFS**.
* It provides increased throughput.
* It provides less average response time and waiting time.

## **Disadvantages-**

* There is an overhead of finding out the closest request.
* The requests which are far from the head might starve for the CPU.
* It provides high variance in response time and waiting time.
* Switching the direction of head frequently slows down the algorithm.

## **SCAN Disk Scheduling Algorithm-**

* As the name suggests, this algorithm scans all the cylinders of the disk back and forth.
* Head starts from one end of the disk and move towards the other end servicing all the requests in between.
* After reaching the other end, head reverses its direction and move towards the starting end servicing all the requests in between.
* The same process repeats.

## **NOTE-**

* SCAN Algorithm is also called as **Elevator Algorithm.**
* This is because its working resembles the working of an elevator.

## **Advantages-**

* It is simple, easy to understand and implement.
* It does not lead to starvation.
* It provides low variance in response time and waiting time.

## **Disadvantages-**

* It causes long waiting time for the cylinders just visited by the head.
* It causes the head to move till the end of the disk even if there are no requests to be serviced.

## **C-SCAN Disk Scheduling Algorithm-**

* Circular-SCAN Algorithm is an improved version of the **SCAN Algorithm**.
* Head starts from one end of the disk and move towards the other end servicing all the requests in between.
* After reaching the other end, head reverses its direction.
* It then returns to the starting end without servicing any request in between.
* The same process repeats.

## **Advantages-**

* The waiting time for the cylinders just visited by the head is reduced as compared to the SCAN Algorithm.
* It provides uniform waiting time.
* It provides better response time.

## **Disadvantages-**

* It causes more seek movements as compared to SCAN Algorithm.
* It causes the head to move till the end of the disk even if there are no requests to be serviced.

## **LOOK Disk Scheduling Algorithm-**

* LOOK Algorithm is an improved version of the **SCAN Algorithm**.
* Head starts from the first request at one end of the disk and moves towards the last request at the other end servicing all the requests in between.
* After reaching the last request at the other end, head reverses its direction.
* It then returns to the first request at the starting end servicing all the requests in between.
* The same process repeats.

## **NOTE-**

The main difference between SCAN Algorithm and LOOK Algorithm is-

* SCAN Algorithm scans all the cylinders of the disk starting from one end to the other end even if there are no requests at the ends.
* LOOK Algorithm scans all the cylinders of the disk starting from the first request at one end to the last request at the other end.

## **Advantages-**

* It does not causes the head to move till the ends of the disk when there are no requests to be serviced.
* It provides better performance as compared to SCAN Algorithm.
* It does not lead to starvation.
* It provides low variance in response time and waiting time.

## **Disadvantages-**

* There is an overhead of finding the end requests.
* It causes long waiting time for the cylinders just visited by the head.

## **C-LOOK Disk Scheduling Algorithm-**

* Circular-LOOK Algorithm is an improved version of the **LOOK Algorithm**.
* Head starts from the first request at one end of the disk and moves towards the last request at the other end servicing all the requests in between.
* After reaching the last request at the other end, head reverses its direction.
* It then returns to the first request at the starting end without servicing any request in between.
* The same process repeats.

## **Advantages-**

* It does not causes the head to move till the ends of the disk when there are no requests to be serviced.
* It reduces the waiting time for the cylinders just visited by the head.
* It provides better performance as compared to LOOK Algorithm.
* It does not lead to starvation.
* It provides low variance in response time and waiting time.

## **Disadvantages-**

* There is an overhead of finding the end requests.