

Relatively speaking, retrieving data from hard disk drivers is always slow compared to CPU and memory access due to the mechanical nature of the magnetic disk. Disk arm movement is very expensive operation therefore operating systems use disk scheduling algorithms to reduce seek time. Below you can find a summary of some of the well known disk scheduling algorithms. The algorithm receives a queue of request positions (track numbers) and the current head position. The output of the algorithm is the order in which the requests are serviced.

### **First Come First Serve (FCFS)**

1.
  1. Requests are serviced in the order in which they arrive
  2. The algorithm is easy to implement
  3. Bad algorithm as it may involve lots of unnecessary seek distance

Here is an example assuming we have 200 cylinders:

```
1 Queue: 100, 175, 51, 133, 8, 140, 73, 77
2 Head: 63
3 Seek order: 100, 175, 51, 133, 8, 140, 73, 77
4 Seek distance = (100-63) + (175-100) + (175-51) + (133-51) +
5               (133-8) + (140-8) + (140-73) + (77-73)
6               = 646
```

### **Shortest Seek Time First (SSTF)**

1.
  1. Service request with the shortest seek time from the current head position
  2. May starve some requests
  3. Good algorithm for a small list of requests

Here is an example:

```
1 Queue: 100, 175, 51, 133, 8, 140, 73, 77
2 Head: 63
3 Seek order: 73, 77, 51, 8, 100, 133, 140, 175
4 Seek distance = 10 + 4 + 26 + 43 + 92 + 33 + 7 + 35 = 250
```

### **Scan or Elevator Algorithm**

1.
  1. Disk arm starts at one end of the disk and moves towards the other end
  2. It services the requests as it moves then reverses the direction when it hits the other end
  3. Scan algorithms are good for heavy loads and more fair

Here is an example:

```
1 Queue: 100, 175, 51, 133, 8, 140, 73, 77
2 Head: 63
3 Seek order: 51, 8, 0, 73, 77, 100, 133, 140, 175
4 Seek distance = 12 + 43 + 8 + 73 + 4 + 23 + 33 + 7 + 35 = 238
```

### **C-Scan**

1.
  1. Similar to scan algorithm but the head returns to cylinder 0 when it reaches the end of the disk
  2. Treats the cylinder list as a circular list that wraps around the disk
  3. Waiting time is more uniform for cylinders near the edge of the disk

Here is an example:

1 Queue: 100, 175, 51, 133, 8, 140, 73, 77  
2 Head: 63  
3 Seek order: 73, 77, 100, 133, 140, 175, 199, 0, 8, 51  
4 Seek distance =  $10 + 4 + 23 + 33 + 7 + 35 + 24 + 199 + 8 + 43$   
5 = 387

### C-Look

1.

1. Similar to C-Scan but the head only travels as far as the last request in each direction

Here is an example:

1 Queue: 100, 175, 51, 133, 8, 140, 73, 77  
2 Head: 63  
3 Seek order: 73, 77, 100, 133, 140, 175, 8, 51  
4 Seek distance =  $10 + 4 + 23 + 33 + 7 + 35 + 167 + 43$   
5 = 322

### Summary

Here is a tabular summary of what we have mentioned:

Algorithm	Properties
First Come First Serve (FCFS)	Requests are serviced in the order in which they arrive Easy to implement May involve unnecessary seek distance
Shortest Seek Time First (SSTF)	Service request with the shortest seek time from the current head position May starve some requests Good for a small list of requests
Scan or Elevator Algorithm	Disk arm starts at one end of the disk and moves towards the other end It services the requests as it moves then reverses the direction when it hits the other end Good for heavy loads and more fair
C-Scan	Head returns to cylinder 0 when it reaches the end of the disk Treats the cylinder list as a circular list that wraps around the disk Waiting time is more uniform for cylinders near the edge of the disk
C-Look	Similar to C-Scan but the head only travels as far as the last request in each direction

Please use the comments section below for questions, corrections or feedback. Thanks for reading.