ISSN: 2321-7782

Volume 1, Issue 2, July 2013

International Journal of Advance Research in Computer Science and Management Studies

Research Paper

Available online at: www.ijarcsms.com

Disk Scheduling

Hetal Paida

Lecturer in G.E.C.

Gujarat Technological University

India

Abstract: The performance of a computer system is to a large extent dependent upon how fast a disk request is serviced. Since most jobs nearly depend upon the disk for input and output purpose. In multi-programmed environment, many processes may be generating requests for reading and writing disk records because these processes often make requests faster than they can be serviced by disk system, writing lines or queues build up for each device.

To service a request, a disk system requires that the head be moved to the desired track, then a wait for latency and finally the transfer of data.

In the paper we will examine several scheduling algorithms.

Keywords: Seek time, Rotational latency, FCFS, SSTF, SCAN and C-SCAN.

I. Introduction

The disk is a resource which has to be shared. It is therefore has to be scheduled for use, according to some kind of the scheduling system. The secondary storage media structure is one of the vital parts of the file system. Disks are the one, providing lots of the secondary storage. As compared to magnetic tapes, disks have very fast access time and disk bandwidth. The access time has two major constituents: seek time and the rotational latency.

The seek ***time*** is the time required for the disk arm to move the head to the cylinder with the desired sector. The ***rotational latency*** is the time required for the disk to rotate the desired sector until it is under the read-write head. The ***disk bandwidth*** is the total number of bytes transferred per unit time.

Disk drivers are large single dimensional arrays of logical blocks to be transferred. Because of large usage of disks, proper scheduling algorithms are required. A scheduling policy should attempt to maximize throughput and also to minimize mean response time.

Scheduling Algorithms

In this paper we have considered the four types of scheduling algorithms.

(1) FCFS Scheduling

(2) SSTF Scheduling

(3) SCAN Scheduling

(4) C-SCAN Scheduling

II. First-Cum-First-Served (FCFS) Scheduling

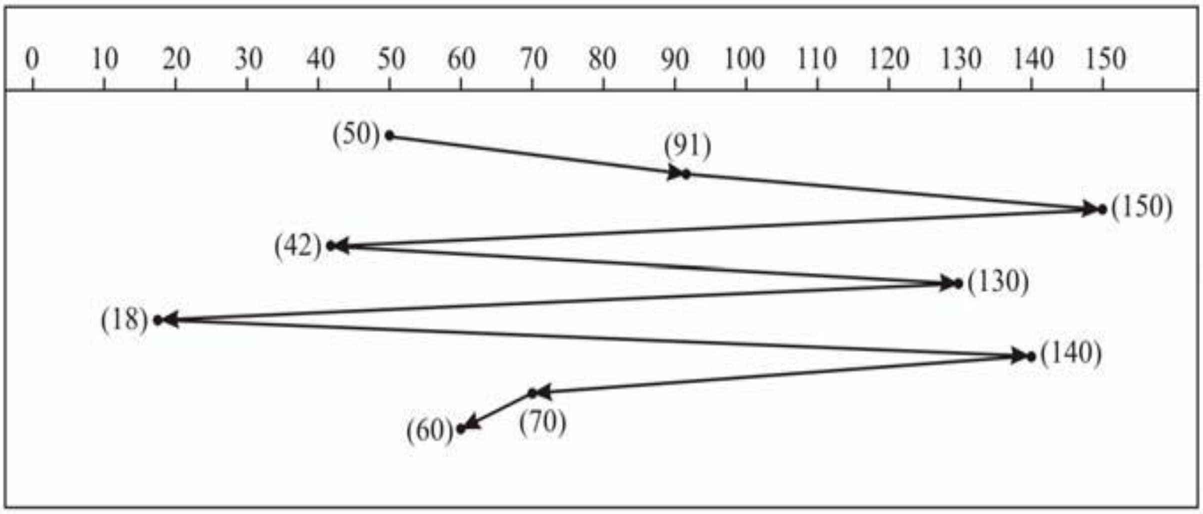
This is the simplest form of disk scheduling in which the first request to arrive is the first one serviced.

FCFS has a fair policy in the sense that once a request has arrived, its place in the schedule is fixed irrespective of arrival of a higher priority request.

FCFS is easy to program, however, it may not provide the best service. Suppose the requests for inputting/outputting to blocks on the cylinders have arrived, forming the following disk queue:

50 ,91, 150 ,42, 130, 18, 140, 70 ,60

Also assume that the disk head is initially at cylinder 50 then it moves to 91, then to150 and so on. The total head movement in this scheme is 610 cylinders, which makes the system slow because of wild swings. Proper scheduling while moving towards a particular direction could decrease this. This will further improve performance. FCFS scheme is clearly represented in below Figure.



III. Shortest-Seek-Time-First (SSTF) Scheduling

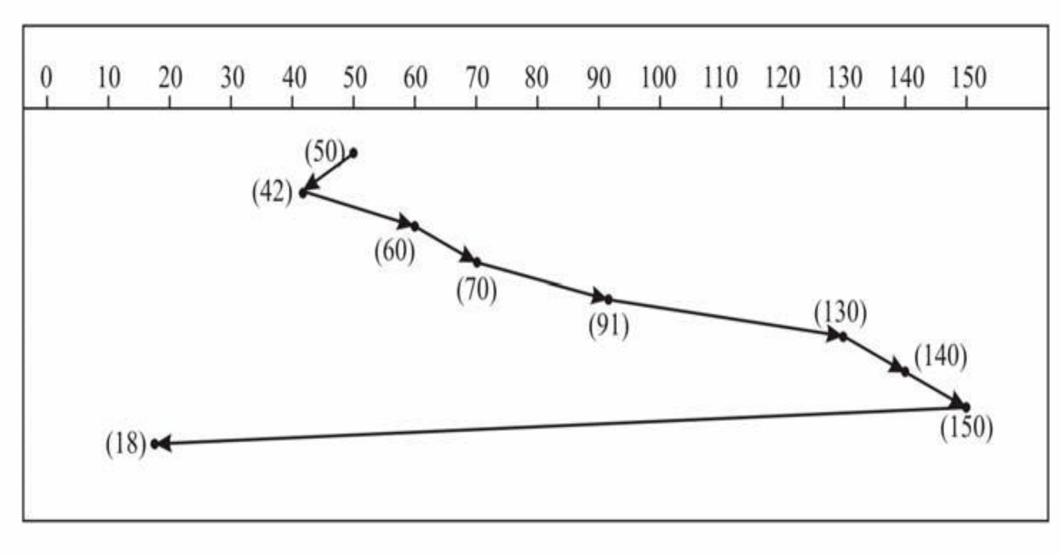
In Shortest-Seek-Time-First (SSTF) Scheduling priority is given to those processes which need the shortest seek, even if these requests are not the first ones in the queue.

It means that all requests are not the nearer to the current head position are serviced together before moving head to distance tracks. Since seek time is generally proportional to the track difference between the requests; this approach is implemented by moving the head to the closest track in the request queue.

In the previous disk queue sample the cylinder close to critical head position i.e., 50, is 42 cylinders, next closest request is at 60. From there, the closest one is 70, then 91,130,140,150 and then finally 18-cylinder. This scheme has reduced the total head movements to 248 cylinders and hence improved the performance.

Like SJF (Shortest Job First) for the CPU scheduling SSTF also suffers from starvation problem. This is because requests may arrive at any time. Suppose we have the requests in the disk queue for cylinders 18 and 150, and while servicing the 18-cylinder request, some other request closest to it arrives and it will be serviced next. This can continue further also making the request at 150-cylinder wait for long. Thus a continual stream of requests near one another could arrive and keep the far away request waiting indefinitely.

The SSTF is not the optimal scheduling due to the starvation problem. This whole scheduling is shown in below Figure.



IV. SCAN scheduling

Early

A SCAN scheduling strategy was developed to overcome the discrimination as well as to recognize the dynamic nature of the request queue.

The read/write head of a disk start from one end and move towards the other end, service requests as it reaches each track until it reaches to the other end of the disk.

After reaching another end of disk, disk head reverses its path-direction while continuing with services whichever comes on the way. This way disk head continuously oscillates from end to end.

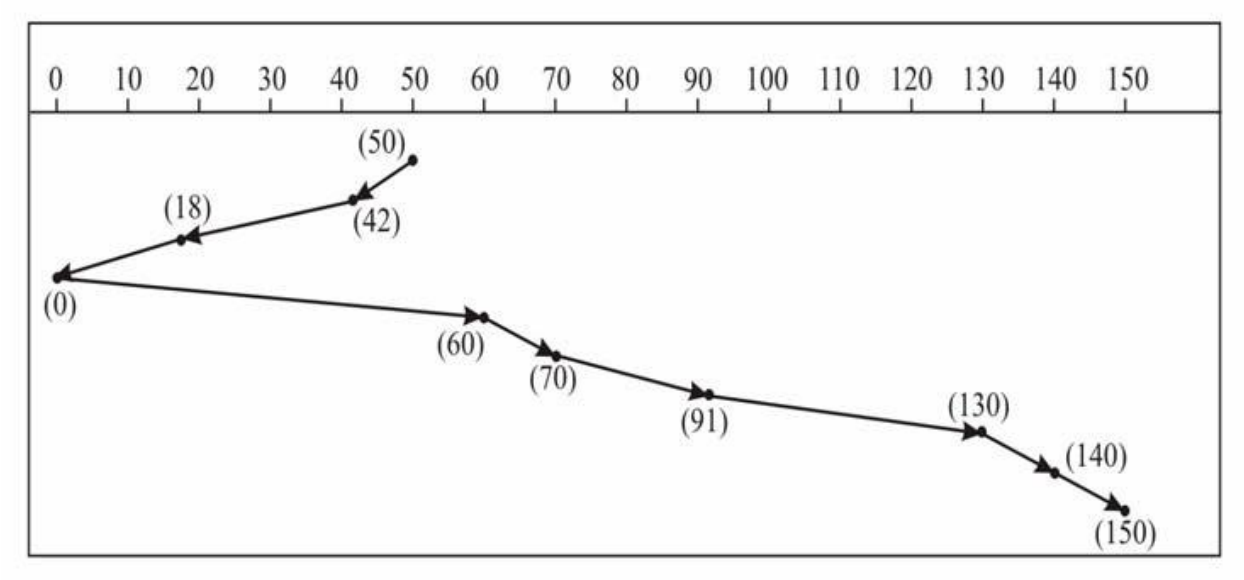
In the example problem two things must be known before starting the scanning process. Firstly, the initial head position i.e., 50 and then the head movement direction (let it towards 0, starting cylinder). Consider the disk queue again:

91, 150, 42, 130, 18, 140, 70, 60

Starting from 50 it will move towards 0, servicing requests 42 and 18 in between. At cylinder 0 the direction is reversed and the arm will move towards the other end of the disk servicing the requests at 60, 70, 91,130,140 and then finally 150.

As the arm acts like an elevator in a building, the SCAN algorithm is also known as elevator algorithm sometimes. The limitation of this scheme is that few requests need to wait for a long time because of the reversal of head direction. This scheduling algorithm results in a total head movement of only 200 cylinders.

Volume 1, Issue 2, July 2013 pg. 6-9



V. C-SCAN Schudling

Similar to SCAN algorithm, C-SCAN also moves head from one end to the other servicing all the requests in its way. The difference here is that after the head reaches the end it immediately returns to the beginning, skipping all the requests on the return trip.

The servicing of the requests is done only along one path. Thus comparatively this scheme gives uniform wait time because cylinders are like circular lists that wrap around from the last cylinder to the first one.

VI. Conclusion

The aspect of Input-Output that has the greatest impact on overall system performance is disk I/O. Accordingly; there has been greater research and design effort in this area than in any other kind of I/O. One of the most widely used approaches to improve disk Input- Output performance is Disk Scheduling.

At any time, there may be a queue of requests for I/O on the same disk. It is the object of disk scheduling to satisfy these requests in a way that minimizes the mechanical seek time of the disk and hence improves performance.

References

Books:

1. Willing Stallings. Operating System Internals and Design Principles-Third edition. New Jersey: Prentice-Hall International.
2. Dr. Arvind Mohan Parashar, Chandresh Shah, Suburb Mishra, Sunilkumar Ojha Computer Science & Application. Agra: Upkar Prakashan.
3. Peter B. Galvin, Abraham Silberschatz, Gerg Gagne – Operating System Concepts 8th Edition, Wiley.
4. Andrew S. Tanenbaum, Modern Operating system Concept 3rd Edition, PHI Learning.
5. Dhananjay M. Dhamdhere, Operating System: A Concept Based Approach 3rd Edition, Tata McGraw - Hill Education
6. Shubhra Garg, Fundamentals of Distributed Operating System – 2010, S. K. Kataria & Sons.

ISSN: 2321-7782