Disk Scheduling Algorithm

First Come First Serve (FCFS)

It is the simplest form of disk scheduling algorithms. The I/O requests are served or processed according to their arrival. The request arrives first and will be accessed and served first.

Example: Given the following track requests in the disk queue, compute for the Total Head Movement2 (THM) of the read/write head: 95, 180, 34, 119, 11, 123, 62, 64 .

Consider that the read/write head is positioned at location 50. Prior to this track location 199 was serviced. Show the total head movement for a 200 track disk (0-199).
Solution:
Total Head Movement Computation:
(THM) = (180 - 50) + (180-34) + (119-34) + (119-11) + (123-11) + (123-62) + (64-62)
= 130 + 146 + 85 + 108 + 112 + 61 + 2
(THM) = 644
tracks Assuming a seek rate of 5 milliseconds is given, we compute for the seek time using the formula:
Seek Time = THM * Seek rate
= 644 * 5 ms
Seek Time = 3,220 ms

There are some requests that are far from the current location of the R/W head which causes the access arm to travel from innermost to the outermost tracks of the disk or vice versa. In this example, it had a total of 644 tracks and a seek time of 3,220 milliseconds. Based on the result, this algorithm produced a higher seek rate since it follows the arrival of the track requests.

Shortest Seek Time First (SSTF)

This algorithm is based on the idea that the R/W head should proceed to the track that is closest to its current position. The process would continue until all the track requests are taken care of.

Example: Given the following track requests in the disk queue, compute for the Total Head Movement2 (THM) of the read/write head: 95, 180, 34, 119, 11, 123, 62, 64

Consider that the read/write head is positioned at location 50. Prior to this track location 199 was serviced. Show the total head movement for a 200 track disk (0-199).

Solution:
(THM) = 12+2+30+23+84+24+4+57
(THM) = 236 tracks

Seek Time = THM * Seek rate
= 236 * 5ms
Seek Time = 1,180 ms

In this algorithm, request is serviced according to the next shortest distance. Starting at 50, the next shortest distance would be 62 instead of 34 since it is only 12 tracks away from 62 and 16 tracks away from 34. The process would continue up to the last track request. There are a total of 236 tracks and a seek time of 1,180 ms, which seems to be a better service compared with FCFS .

SCAN

This algorithm is performed by moving the R/W head back-and-forth to the innermost and outermost track. As it scans the tracks from end to end .

Example: Given the following track requests in the disk queue, compute for the Total Head Movement2 (THM) of the read/write head: 95, 180, 34, 119, 11, 123, 62, 64

Solution:
(THM) = (50-0) + (180-0) = 50 + 180

(THM) = 230
Seek Time = THM * Seek rate
= 230 * 5ms
Seek Time = 1,150 ms
This algorithm works like an elevator does.

In the algorithm example, it scans down towards the nearest end and when it reaches the bottom it scans up servicing the requests that it did not get going down. If a request comes in after it has been scanned, it will not be serviced until the process comes back down or moves back up. This process moved a total of 230 tracks and a seek time of 1,150. This is optimal than the previous algorithm.

Look

The disk arm starts at the first I/O request on the disk, and moves toward the last I/O request on the other end, servicing requests until it gets to the other extreme I/O request on the disk, where the head movement is reversed and servicing continues. It moves in both directions until both last I/O requests; more inclined to serve the middle cylinder requests.

Example: Given the following track requests in the disk queue, compute for the Total Head Movement2 (THM) of the read/write head: 95, 180, 34, 119, 11, 123, 62, 64 ALI
Solution:
(THM) = (50-11) + (180-11) = 39 + 169
(THM) = 208 tracks

Seek Time = THM * Seek rate
= 208 * 5ms
Seek Time = 1,040 ms

This algorithm has a result of 208 tracks and a seek rate of 1,040 milliseconds. This algorithm is better than the previous algorithm.

C-Scan

The head moves from one end of the disk to the other, servicing requests as it goes. When it reaches the other end, however, it immediately returns to the beginning of the disk, without servicing any requests on the return trip.
Treats the cylinders as a circular list that wraps around from the last cylinder to the first one.
Provides a more uniform wait time than SCAN; it treats all cylinders in the same manner.

C-Look

Look version of C-Scan.
Arm only goes as far as the last request in each direction, then reverses direction immediately, without first going all the way to the end of the disk.
In general, Circular versions are more fair but pay with a larger total seek time.
Scan versions have a larger total seek time than the corresponding Look versions.

RSS

It stands for random scheduling and just like its name it is nature. It is used in situations where scheduling involves random attributes such as random processing time, random due dates, random weights, and stochastic machine breakdowns this algorithm sits perfect. Which is why it is usually used for analysis and simulation.

LIFO

In LIFO (Last In, First Out) algorithm, newest jobs are serviced before the existing ones i.e. in order of requests that get serviced the job that is latest or last entered is serviced first and then the rest in the same order.
Advantages

- Maximizes locality and resource utilization

Disadvantages

- Can seem a little unfair to other requests and if new requests keep coming in, it causes starvation to the old and existing ones.

N-STEP SCAN

It is also known as N-STEP LOOK algorithm. In this a buffer is created for N requests. All requests belonging to a buffer will be serviced in one go. Also once the buffer is full no new requests are kept in this buffer and are sent to another one. Now, when these N requests are serviced, the time comes for another top N requests and this way all get requests get a guaranteed service

Advantages

It eliminates starvation of requests completely

FSCAN

This algorithm uses two sub-queues. During the scan all requests in the first queue are serviced and the new incoming requests are added to the second queue. All new requests are kept on halt until the existing requests in the first queue are serviced.

Advantages

FSCAN along with N-Step-SCAN prevents “arm stickiness” (phenomena in I/O scheduling where the scheduling algorithm continues to service requests at or near the current sector and thus prevents any seeking)