Disk Scheduling

When a process needs I/O to or from the disk it issue a system call to the OS. This request specifies several pieces of information:

1. whether this operation is input or output
2. What is the disk address
3. What is the memory address
4. What is the no. of bytes to be transfer.

If the desired disk drive and disk controller are available, the request can be serviced immediately. If the driver and controller are busy, new request can be placed on the queue of pending request.

For multiprogramming system with many processes, the disk queue may have several pending requests. When one request is completed, the OS chooses by the scheduler which pending request requires to be required next.

For this several disk scheduling algorithms are differed which are used in accessing and transferring the data.

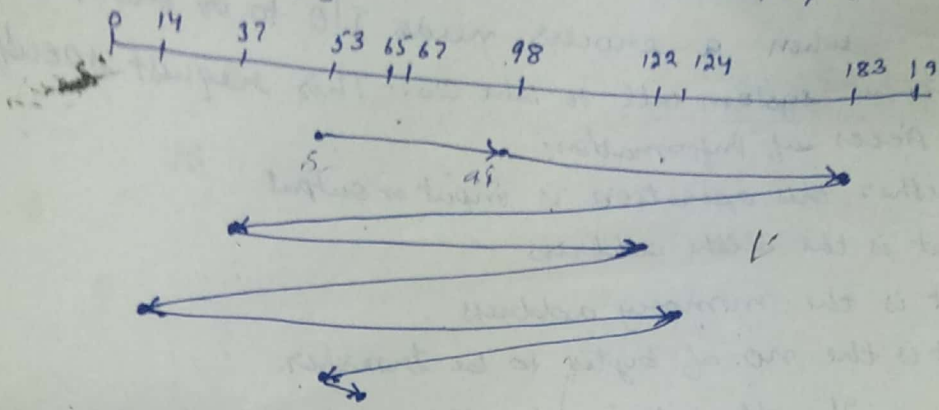
1. **FCFS** :- It is a simplest form of disk scheduling i.e first come, first served algorithm. It does not provide the fastest service.

   e.g:- a disk queue with requests for I/O to blocks on tracks

   98, 183, 37, 122, 14, 124, 65, 67

   in that order. If disk head initially at track 53, it will first move from 53 to 98 then to 183, 37, 122, 14, 124, 65 and finally to 67 for a total head movement of 640 tracks. The problem with this schedule is it will swing from 122 to 14 and then back to 124. If the requests for the tracks 37 and 14 could be serviced together, before or after the requests at 122 and 124, the total head movement could be decreased and performance could be improved.

Queue = 98, 183, 37, 122, 14, 124, 65, 67
Head starts at 53



FCFS disk Scheduling

SSTF Scheduling :- (Shortest-seek-time-first) :-

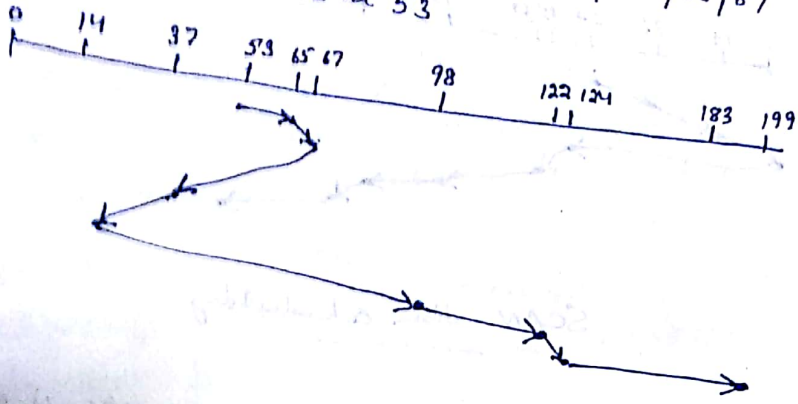
It service all the requests close to the current head position, before moving the head far away to service other requests. The SSTF algorithm selects the request with the minimum seek time from the current head position. SSTF chooses the pending request closest to the current head position.

E.g.; The requested queue, the closest request to the initial head position (53) is at track 65. Once we are at track 65, the next closest is at track 67, From there at track 37 is closer than 98, so 37 is served next. then we service the request at track 14, then 98, 122, 124 and finally 183. The total head movement is 236 tracks.

Like SJF, it may cause starvation of some request. Suppose we have two requests in queue for track 14 & 186, and while servicing the request from 14, a new request near 14 arrives. The new request will be serviced next making the request at 186 wait. During this request for track 186 to wait indefinitely.

Queue = 98, 183, 37, 122, 14, 124, 65, 67

Head starts at 53

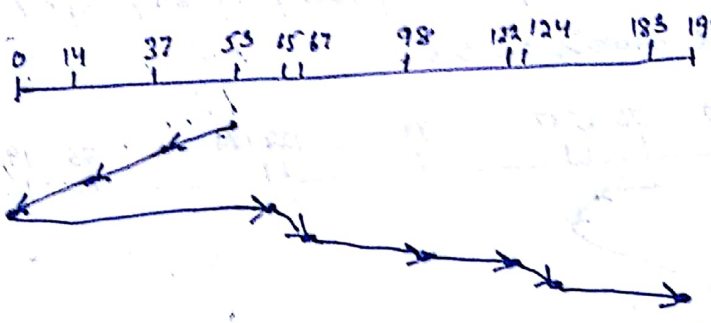


SCAN Scheduling :- In this, the disk arm starts at one end of the disk and moves toward the other end, servicing requests as it reaches each ~~cylinder~~ track, until it gets to the other end of the disk. At the other end the direction of the head movement is reversed and servicing continues. The head continuously scans back and forth across the disk.

Before applying SCAN to schedule the requests on tracks 98, 183, 37, 122, 14, 124, 65 and 67, we need to know the direction of head movement. If the disk arm is moving toward 0, the head will service 37 and then 14. At track 0 the arm ~~is moving toward~~ will reverse and will move toward the other end of the disk servicing the requests at 65, 67, 98, 122, 124 and 183. If a request arrives in the queue just in front of the head, it will be serviced immediately; a request arriving just behind the head will have to wait until the arm moves to the end of the disk, reverses direction and come back.

SCAN algorithm is also called the elevator algorithm

queue = 98, 183, 37, 122, 14, 124, 65, 67

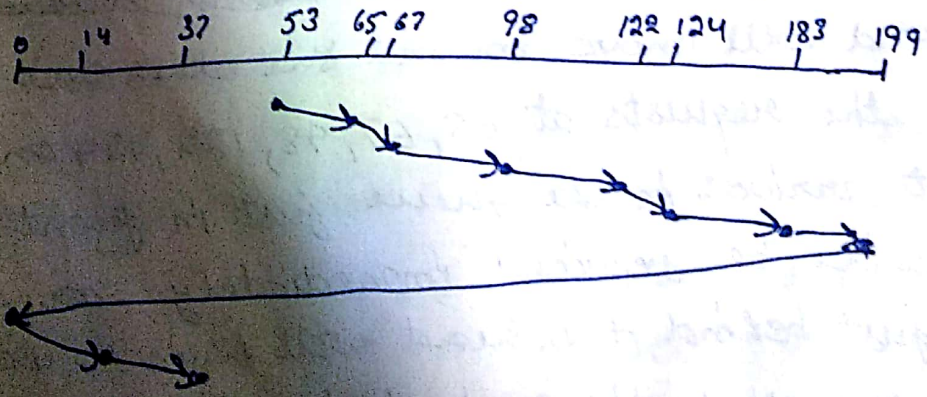


SCAN disk scheduling.

C-SCAN:- The heaviest density of requests is at the other end of the disk. These request have also waited the longest, so why not go there first. That is the idea of this algorithm.

Circular SCAN (C-SCAN) scheduling is a variant of SCAN designed to provide a more uniform wait time. Like SCAN, C-SCAN moves the head from one end of the disk to the other, servicing requests along the way. When the head reaches the other end, it immediately returns to the beginning of the disk, without servicing any requests on the return.

queue = 98, 183, 37, 122, 14, 124, 65, 67

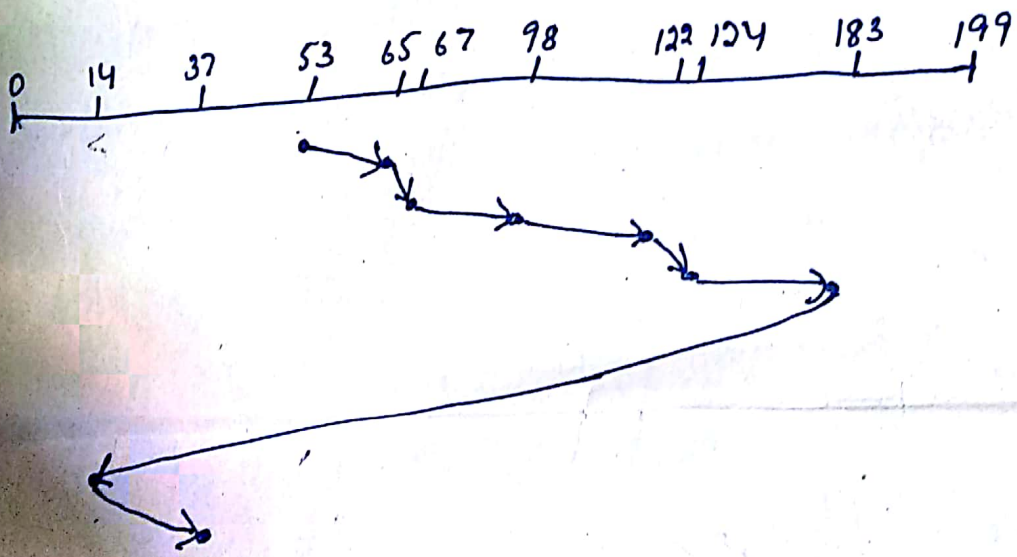


C-SCAN disk Scheduling

Look Scheduling :-

Both SCAN and C-SCAN move the disk arm across the full width of the disk. More commonly the arm goes only as far as the final request in each direction. Then, it reverses direction immediately, without going all the way to the end of the disk. The versions of SCAN and C-SCAN are called LOOK and C-LOOK scheduling, because they look for a request before continuing to move in a given direction.

queue 98, 183, 37, 122, 14, 124, 65, 67



C-LOOK disk Scheduling