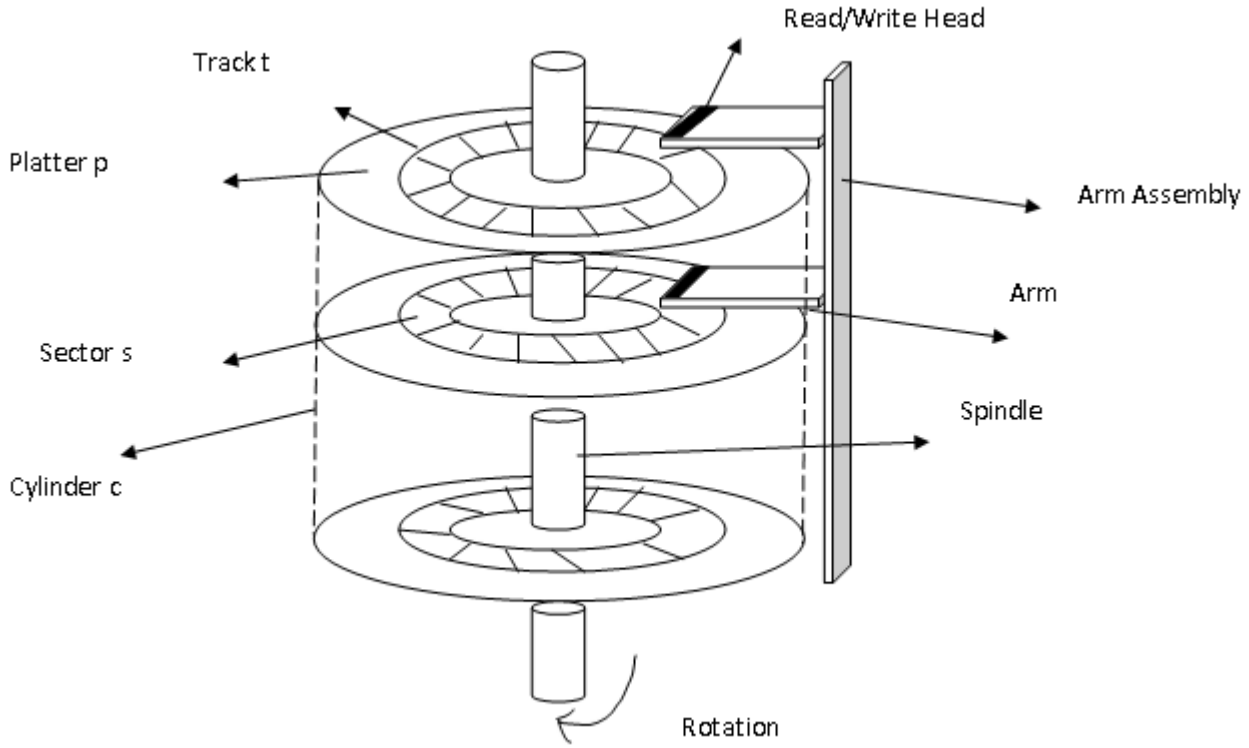
SNS COLLEGE OF TECHNOLOGY

(Autonomous)

COIMBATORE-35

MASS STORAGE STRUCTURE:DISK SCHEDULING

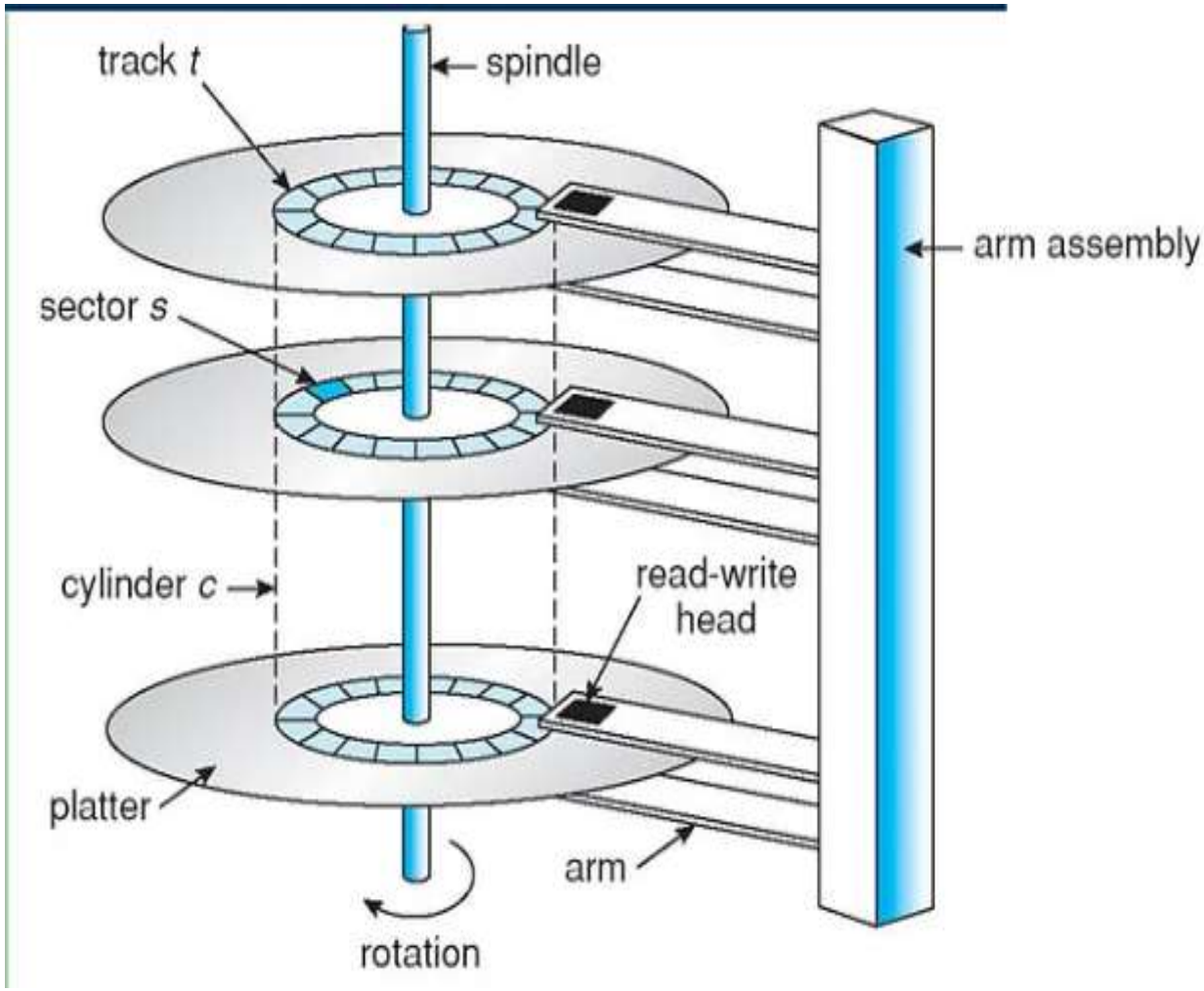


Mass Storage Structure:disk Scheduling

- The operating system is responsible for using hardware efficiently — for the disk drives, this means having a fast access time and disk bandwidth
- Minimize seek time
- Seek time ≈ seek distance
- Disk **bandwidth** is the total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer

Mass Storage Structure:disk Scheduling

- There are many sources of disk I/O request
  - OS
  - System processes
  - Users processes
- I/O request includes input or output mode, disk address, memory address, number of sectors to transfer
- OS maintains queue of requests, per disk or device
- Idle disk can immediately work on I/O request, busy disk means work must queue
  - Optimization algorithms only make sense when a queue exists


track t
spindle
arm assembly
sector s
cylinder c
read-write
head
platter
arm
rotation

Mass Storage Structure:disk Scheduling

- Note that drive controllers have small buffers and can manage a queue of I/O requests (of varying “depth”)
- Several algorithms exist to schedule the servicing of disk I/O requests
- The analysis is true for one or many platters
- We illustrate scheduling algorithms with a request queue (0-199)

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

Head pointer 53

Mass Storage Structure:disk Scheduling

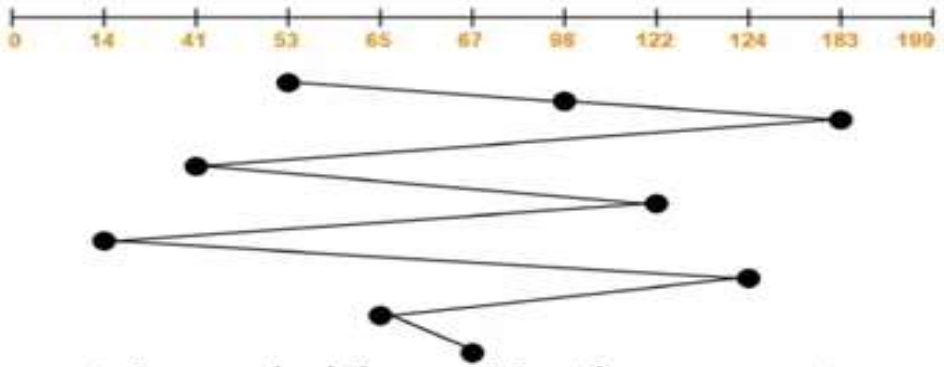
FCFS

Problem - FCFS

Consider a disk queue with requests for I/O to blocks on cylinders **98, 183, 41, 122, 14, 124, 65, 67**. The head is initially at cylinder number 53. The cylinders are numbered from 0 to 199. The total head movement in number of cylinders incurred while servicing these requests is _

Solution-

Order of head movements = 53 -> 98 -> 183 -> 41 -> 122 -> 14 -> 124 -> 65 -> 67



Total head movements incurred while servicing these requests

= |53 - 98| + |98 - 183| + |183 - 41| + |41 - 122| + |122 - 14| + |14 - 124| + |124 - 65| + |65 - 67| = 45 + 85 + 142 + 81 + 108 + 110 + 59 + 2 = **632**

Mass Storage Structure:disk Scheduling

SSTF

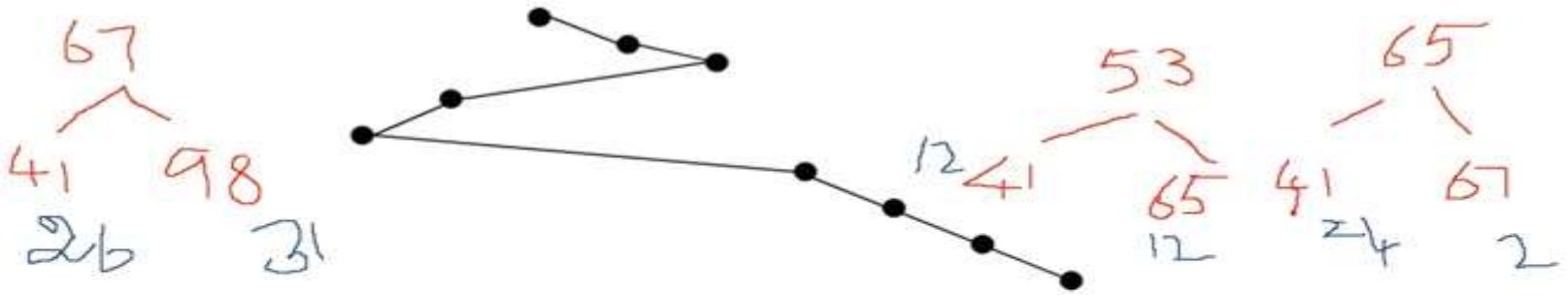
- Shortest Seek Time First selects the request with the minimum seek time from the current head position

Problem: SSTF

Disk queue with requests for I/O to blocks on cylinders **98, 183, 41, 122, 14, 124, 65, 67**. The head is initially at cylinder number 53 and moving towards the end. The cylinders are numbered from 0 to 199. The total head movement in number of cylinders incurred while servicing these requests is _

Solution-

Order of head movements = 53 -> 65 -> 67 -> 41 -> 14 -> 98 -> 122 -> 124 -> 183



Total head movements incurred while servicing these requests

= |53 - 65| + 165 - 67| + |67 - 41| + |41 - 14| + |14 - 98| + |98 - 122| + |122 - 124| + |124 - 183| = 12 + 2 + 26 + 27 + 84 + 24 + 2 + 59 = **236**

SCAN

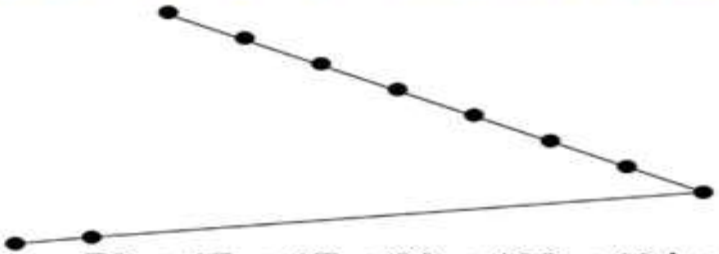
- The disk arm starts at one end of the disk, and moves toward the other end, servicing requests until it gets to the other end of the disk, where the head movement is reversed and servicing continues.
- **SCAN algorithm** Sometimes called the **elevator algorithm**
- Illustration shows total head movement of 236 cylinders
- But note that if requests are uniformly dense, largest density at other end of disk and those wait the longest

SCAN (Cont.)

Problem- SCAN

Disk queue with requests for I/O to blocks on cylinders **98, 183, 41, 122, 14, 124, 65, 67**. The head is initially at cylinder number 53 and moving towards the end. The cylinders are numbered from 0 to 199. The total head movement in number of cylinders incurred while servicing these requests is _

Solution-



Order of head movements = 53 -> 65 -> 67 -> 98 -> 122 -> 124 -> 183 -> 199 -> 41 -> 14

Total head movements incurred while servicing these requests

= |53 - 65| + 165 - 67| + |67 - 98| + |98 - 122| + |122 - 124| + |124 - 183| + |183 - 199| + |199 - 41| + |41 - 14|

= 12 + 2 + 31 + 24 + 2 + 59 + 16 + 158 + 27 = 331

Verification: = |199 - 53| + |199 - 14| = 146 + 185 = 331

C-SCAN

C-SCAN Disk Scheduling Algorithm-

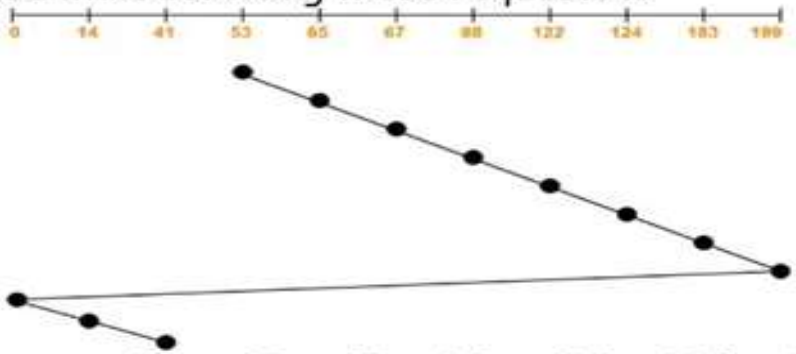
- Circular-SCAN Algorithm is an improved version of SCAN algorithm.
- Disk head - start from one end of the disk and move towards the other end of the disk servicing all the requests in between.
- After reaching the other end, head reverses its direction.
- It then returns to the starting end without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

C-SCAN

Problem- C-SCAN

Disk queue with requests for I/O to blocks on cylinders **98, 183, 41, 122, 14, 124, 65, 67**. The head is initially at cylinder number 53 and moving towards the end. The cylinders are numbered from 0 to 199. The total head movement in number of cylinders incurred while servicing these requests is

Solution-



Order of head movements=53 -> 65 -> 67 -> 98 -> 122 -> 124 -> 183 -> 199 -> 0 -> 14 -> 41

Total head movements incurred while servicing these requests

= |53 - 65| + |65 - 67| + |67 - 98| + |98 - 122| + |122 - 124| + |124 - 183| + |183 - 199| + |199 - 0| + |0 - 14| + 114 - 41|

= 12 + 2 + 31 + 24 + 2 + 59 + 16 + 199 + 14 + 27 = 386

Verification: = |199 - 53| + |199 - 0| + |41 - 0| = 146 + 199 + 41 =386

LOOK

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 is repeated

NOTE-

The main difference between SCAN Algorithm and LOOK Algorithm is-

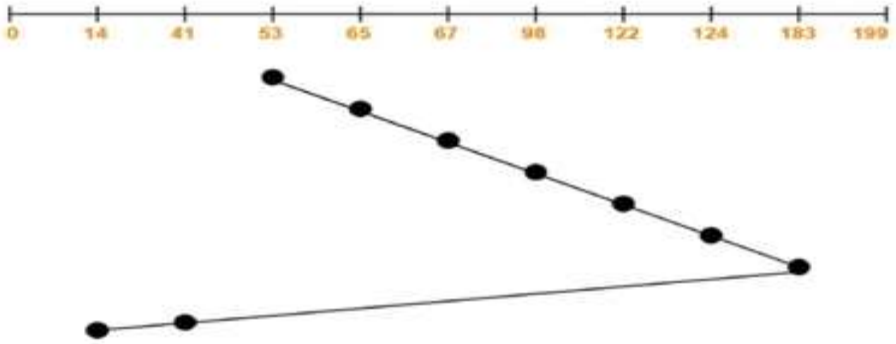
- SCAN Algorithm scans all the cylinders of the disk starting from one end to the other end even if there are no requests at the ends.
- LOOK Algorithm scans all the cylinders of the disk starting from the first request at one end to the last request at the other end.

LOOK

Problem-LOOK

Disk queue with requests for I/O to blocks on cylinders **98, 183, 41, 122, 14, 124, 65, 67**. The head is initially at cylinder number 53 and moving towards the end. The cylinders are numbered from 0 to 199. The total head movement in number of cylinders incurred while servicing these requests is _

Solution-



Order of head movements = 53 -> 65 -> 67 -> 98 -> 122 -> 124 -> 183 -> 41 -> 14

Total head movements incurred while servicing these requests

= |53 - 65| + |65 - 67| + |67 - 98| + |98 - 122| + |122 - 124| + |124 - 183| + |183 - 41| + |41 - 14| = 12 + 2 + 31 + 24 + 2 + 59 + 142 + 27 = 299

Verification: = |183 - 53| + |183 - 14| = 130 + 169 =299

C -LOOK

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 is repeated

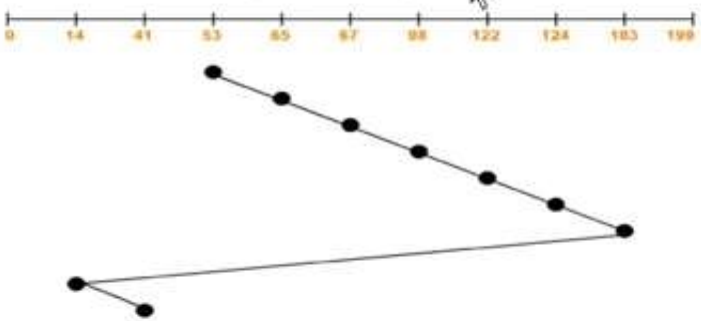
Mass Storage Structure:disk Scheduling

C -LOOK

Problem – C-LOOK

Disk queue with requests for I/O to blocks on cylinders **98, 183, 41, 122, 14, 124, 65, 67**. The head is initially at cylinder number 53 and moving towards the end. The cylinders are numbered from 0 to 199. The total head movement in number of cylinders incurred while servicing these requests is _

Solution-



Order of head movements = 53 -> 65 -> 67 -> 98 -> 122 -> 124 -> 183 -> 14 -> 41

Total head movements incurred while servicing these requests

= |53 - 65| + |65 - 67| + |67 - 98| + |98 - 122| + |122 - 124| + |124 - 183| + |183 - 14| + |14 – 41| = 12 + 2 + 31 + 24 + 2 + 59 + 169 + 27 = 326

Verification: = |183 - 53| + |183 - 14| + |41 - 14| = 130 + 169 + 27 = 326

Mass Storage Structure:disk Scheduling

Selecting a Disk-Scheduling Algorithm

- SSTF is common and has a natural appeal
- SCAN and C-SCAN perform better for systems that place a heavy load on the disk
  - Less starvation
- Performance depends on the number and types of requests
- Requests for disk service can be influenced by the file-allocation method
  - And metadata layout
- The disk-scheduling algorithm should be written as a separate module of the operating system, allowing it to be replaced with a different algorithm if necessary
- Either SSTF or LOOK is a reasonable choice for the default algorithm
- What about rotational latency?
  - Difficult for OS to calculate
- How does disk-based queueing effect OS queue ordering efforts?

References

1. Silberschatz, Galvin, and Gagne, “Operating System Concepts”, Ninth Edition, Wiley India Pvt Ltd, 2009.
2 . Andrew S. Tanenbaum, “Modern Operating Systems”, Fourth Edition, Pearson Education, 2010.

Summarization