

Operating Systems
Course Code: **71203002004**
Disk Scheduling Algorithms

*by -
Minal Rajwar*



Disk Scheduling Algorithms

Disk scheduling (or I/O scheduling) decides the order in which read/write requests to a disk are processed.

The goal is to improve disk performance by reducing access time and increasing efficiency.

Why Disk Scheduling is Needed

- Many processes may send I/O requests at the same time, but the disk can handle only one request at a time.
- Requests may be scattered on different tracks, increasing seek time.
- Since hard disks are slower than memory, we must schedule requests efficiently to improve system speed.

Important Terms

Term	Meaning
Seek Time	Time taken to move the read/write head to the track where data is located.
Rotational Latency	Time taken for the desired sector to come under the head.
Transfer Time	Time taken to actually read or write data.
Disk Access Time	Seek Time + Rotational Latency + Transfer Time
Response Time	Time a request waits before starting service.

Goals of Disk Scheduling

- Minimize Seek Time
- Maximize Throughput (requests served per second)
- Minimize Latency (delay before request starts)
- Ensure Fairness (no starvation)
- Efficient Resource Use

Disk Scheduling Algorithms

a) FCFS (First-Come, First-Served)

- Requests are handled in the order they arrive.
- Simple and fair, but not efficient (large head movement possible).

✓ *Example:* If requests are 10, 22, 20, 2 → service in that order.

b) SSTF (Shortest Seek Time First)

- Selects the nearest track to the current head position.
- Reduces average seek time.
- May cause starvation for distant requests.

✓ *Example:* If head is at 50, and requests at 40, 60, 10 → serves 40 first.

Disk Scheduling Algorithms

c) SCAN (Elevator Algorithm)

- The head moves in one direction, servicing all requests, then reverses direction.
- Avoids starvation and works well under heavy load.
- Average waiting time can be higher than SSTF.

✓ *Analogy:* Like an elevator that goes up and down serving floors.

d) C-SCAN (Circular SCAN)

- The head moves in one direction only.
- After reaching the end, it returns quickly to the start without servicing requests backward.
- Provides uniform waiting time.

Disk Scheduling Algorithms

e) LOOK and C-LOOK

- **LOOK:** Similar to SCAN, but the head goes only up to the last request, not to the end of the disk.
- **C-LOOK:** Similar to C-SCAN, but returns to the first request instead of the start of the disk.
- More efficient as unnecessary travel is avoided.

Performance Comparison - Scheduling Algorithms

Algorithm	Advantages	Disadvantages
FCFS	Simple, fair	High average seek time
SSTF	Low average seek time	Starvation possible
SCAN	Avoids starvation	Direction reversal overhead
C-SCAN	Uniform response time	Slightly longer total travel
LOOK	Reduced head movement	More complex to implement
C-LOOK	Efficient and uniform	Complex implementation

Selection Criteria

- **Workload:** Number and pattern of I/O requests.
- **System Requirements:**
 - Fast response → SSTF
 - High throughput → SCAN / C-SCAN
- **Fairness & Starvation Avoidance:** SCAN or C-SCAN preferred.
- **Implementation Simplicity:** FCFS easiest.
- **Performance Goals:** Choose algorithm based on seek time vs fairness balance.

DISCUSSION & REVISION

1. Which disk scheduling algorithm serves requests in the order they arrive?
2. Which algorithm selects the request closest to the current head position?
3. Which algorithm is also called the Elevator Algorithm?
4. In which algorithm does the head move only in one direction and then jump back?
5. Which algorithm is similar to SCAN but stops at the last request instead of the disk end?

REFERENCES

1. <https://www.geeksforgeeks.org/operating-systems/disk-scheduling-algorithms/>
2. <https://workat.tech/core-cs/tutorial/disk-scheduling-algorithms-in-operating-system-os-o5ahnn6mhh>