**Disk Scheduling Algorithms:**

[**https://workat.tech/core-cs/tutorial/disk-scheduling-algorithms-in-operating-system-os-ope5ahnn6mhh#:~:text=Disk%20Scheduling%20Algorithms%20are%20needed,seek%20time%20may%20increase%20more**](https://workat.tech/core-cs/tutorial/disk-scheduling-algorithms-in-operating-system-os-ope5ahnn6mhh#:~:text=Disk%20Scheduling%20Algorithms%20are%20needed,seek%20time%20may%20increase%20more)**.**

**FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK**

Disk scheduling algorithms are designed to efficiently manage the movement of the disk arm/head to fulfill various I/O requests on a disk. The goal is to minimize the total seek time, which is the time required for the disk arm/head to move to the desired track. Here are explanations for the mentioned disk scheduling algorithms:

1. First-Come-First-Serve (FCFS):

- Requests are serviced in the order they arrive.

- Simple and easy to implement.

- May result in poor performance if there are large differences in seek times between consecutive requests (known as the "convoy effect").

2. Shortest Seek Time First (SSTF):

- The request with the shortest seek time is serviced first.

- Minimizes the total seek time.

- May lead to starvation of some requests if there are always requests with shorter seek times.

3. SCAN (Elevator Algorithm):

- The disk arm scans back and forth across the disk.

- It services requests in the direction it is currently moving until there are no more requests in that direction, then it reverses direction.

- Prevents starvation and provides a more balanced approach.

4. Circular SCAN (C-SCAN):

- Similar to SCAN but only moves in one direction.

- When it reaches the end, it jumps to the other end.

- Reduces the turnaround time for requests at the end of the disk.

5. LOOK:

- Similar to SCAN, but the disk arm does not go all the way to the end of the disk.

- Reverses direction when there are no more requests in the current direction.

- Provides a more efficient approach than SCAN by reducing unnecessary movement to the end of the disk.

6. Circular LOOK (C-LOOK):

- Similar to LOOK but only moves in one direction.

- When it reaches the end, it jumps to the other end.

- Reduces the turnaround time for requests at the end of the disk, similar to C-SCAN.

These algorithms aim to balance the trade-off between fairness, starvation prevention, and minimizing seek time. The choice of algorithm depends on the specific characteristics of the workload and the system's requirements. Some algorithms, like SSTF, offer better performance in terms of seek time, but they may not be fair to all requests. Others, like SCAN and LOOK, provide a more balanced approach. The optimal choice may vary based on the specific use case and workload.