



Department of Computer Science & Engineering

Course Title: Operating System Lab

Course Code: CSE 406

Lab Report No: 05

Lab Report: Priority Scheduling

Submission Date: 13-03-2025

Submitted To:

Atia Rahman Orthi

Lecturer,

Department of CSE, UAP

Submitted By:

Name: Amirul Islam Papon

Reg No: 21201076

Sec: B

First-Come, First-Served (FCFS) Disk Scheduling

Problem Statement: Disk scheduling is essential for optimizing the performance of a computer system by efficiently handling multiple disk requests. The First-Come, First-Served (FCFS) scheduling algorithm processes disk requests in the exact order they arrive. However, due to its sequential nature, FCFS may result in inefficient disk movement and increased seek time. The objective of this study is to analyze the working of FCFS disk scheduling and evaluate its effectiveness.

Steps Involved:

1. **Receive Input:** A queue of disk requests and the initial position of the disk head are provided.
2. **Process Requests Sequentially:** The disk head moves to each requested track in the order of arrival.
3. **Calculate Seek Time:** The seek time for each request is computed as the absolute difference between the current and the next request position.
4. **Total Seek Time Calculation:** The total seek time is obtained by summing up all individual seek times.
5. **Display Results:** The sequence of disk head movements and the total seek time are presented.

Source Code:

```
fcfs_scheduling.py > ...
1  def fcfs_disk_scheduling(requests, head):
2      total_seek_time = 0
3      sequence = [head]
4
5      for request in requests:
6          seek_distance = abs(request - head)
7          total_seek_time += seek_distance
8          head = request
9          sequence.append(head)
10
11     return total_seek_time, sequence
12
13 requests = [60, 100, 30, 150, 20]
14 initial_head = 50
15
16 seek_time, movement_sequence = fcfs_disk_scheduling(requests, initial_head)
17
18 print(f"Total Seek Time: {seek_time}")
19 print(f"Head Movement Sequence: {' -> '.join(map(str, movement_sequence))}")
```

Output:

```
Total Seek Time: 370  
Head Movement Sequence: 50 -> 60 -> 100 -> 30 -> 150 -> 20  
  
[Done] exited with code=0 in 0.08 seconds
```

Discussion & Conclusion: FCFS disk scheduling is the simplest algorithm since it ensures that all requests are processed in the order they arrive, avoiding starvation. However, it does not optimize seek time, which can lead to excessive disk head movement, particularly if requests are scattered across the disk. This inefficiency makes FCFS unsuitable for high-performance systems where minimizing seek time is crucial.

Source Code: https://github.com/Amirul-Islam-Papon/Operating-System/blob/main/fcfs_disk_scheduling.py