UNIVERSITY OF ASIA PACIFIC

Course Title: Operating System Lab
Course Code: CSE 406

Date of Submission: 22th August,2025

Submitted by:

Name: Md.Sarjil Hasan

Reg: 22101168

Sec : D

Submitted to:

Atia Rahman Orthi

Lecturer

University of Asia Pacific

Lab 5:
FCFS Disk Scheduling Algorithm.
Problem Statement:

In modern computer systems, multiple processes often request disk I/O operations simultaneously. Since the disk arm (head) cannot serve all requests at once, a scheduling algorithm is required to decide the order of servicing these requests. The main challenge is to minimize disk head movement, access time, and waiting time while ensuring fairness among all processes. The First-Come First-Served (FCFS) disk scheduling algorithm is one of the simplest methods, where requests are handled in the exact order they arrive in the request queue.

Objective:

The objectives of implementing the FCFS disk scheduling algorithm are:

1. To understand how the disk head movement is managed in FCFS scheduling.
2. To calculate performance metrics such as total head movement and average seek time.
3. To analyze the advantages and disadvantages of FCFS compared to other disk scheduling algorithms.
4. To demonstrate fairness in servicing disk requests.

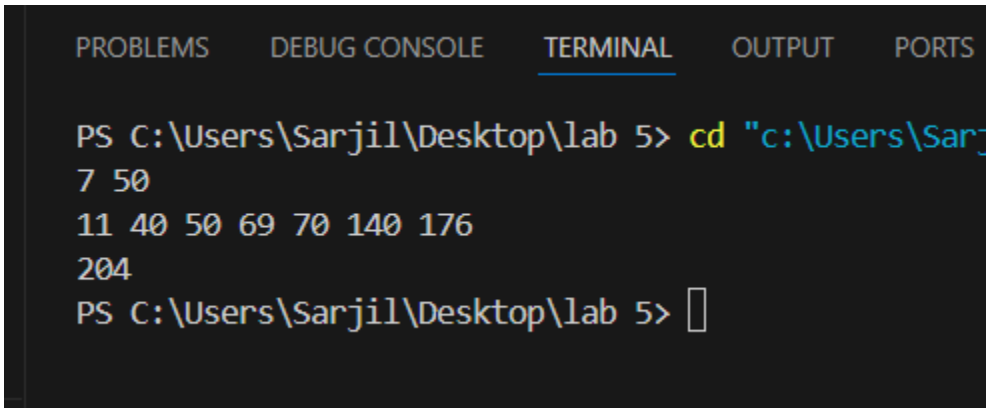
Code:

FCFS_DISK_ALGO.cpp > main()

```cpp
#include <iostream>
#include <vector>
using namespace std;

int main()
{
    int n, head;
    cin >> n >> head;
    int dummy_head=head;
    vector<int> a(n);
    for (int i = 0; i < n; i++)
    {
        cin >> a[i];
    }
    int ans = 0;
    for (int i = 0; i < n; i++)
    {
        if (dummy_head != a[i])
        {
            ans += abs(head - a[i]);
            head = a[i];
        }
    }
    cout << ans << endl;
    return 0;
}
```

Output:



Discussion:

The First-Come First-Served (FCFS) Disk Scheduling Algorithm is the simplest form of disk scheduling, where all I/O requests are handled in the exact order in which they arrive in the queue. The disk controller maintains a request queue, and the disk head moves sequentially to serve each request. This makes the algorithm straightforward and easy to implement.

The main characteristic of FCFS is its fairness: every request is guaranteed to be serviced without the risk of starvation. However, FCFS does not consider the current position of the disk head or the distance between consecutive requests. As a result, the head may travel long distances across the disk, leading to higher seek time and reduced overall efficiency.

While FCFS is predictable and ensures that no process waits indefinitely, it is not optimal for minimizing disk arm movement. In scenarios with widely scattered requests, the disk head may perform excessive movement, causing delays. Therefore, FCFS is often considered suitable for light workloads or when fairness is more important than performance optimization.

Advantages

1. Simplicity – Easy to implement and understand.
2. Fairness – Each request is served in the order it arrives; no starvation occurs.
3. Predictable behavior – Performance can be estimated easily since it follows a straightforward approach.

Disadvantages

1. **Poor average performance** – Large variations in request order may lead to excessive head movement.
2. **High seek time** – Not optimized for minimizing disk head travel.
3. **Inefficient for heavy workloads** – Performs poorly when request queue is large and scattered across the disk.

Conclusion:

The FCFS disk scheduling algorithm provides a simple and fair method for servicing disk I/O requests. However, while it ensures fairness and avoids starvation, it does not minimize head movement or average seek time. Therefore, it is suitable for systems where simplicity and fairness are more important than efficiency. For improved performance in minimizing seek time, algorithms like SSTF, SCAN, or C-SCAN are preferred.