NAME: Meet Raut

DIV: S21

ROLL.NO: 2201084

Experiment 10:

- **AIM: To study and implement the concept of Disk Scheduling.**

- **THEORY:**

As we know, a process needs two type of time, CPU time and IO time. For I/O, it requests the Operating system to access the disk.

However, the operating system must be fare enough to satisfy each request and at the same time, operating system must maintain the efficiency and speed of process execution.
The technique that operating system uses to determine the request which is to be satisfied next is called disk scheduling.

Let's discuss some important terms related to disk scheduling.

Seek Time

Seek time is the time taken in locating the disk arm to a specified track where the read/write request will be satisfied.

Rotational Latency

It is the time taken by the desired sector to rotate itself to the position from where it can access the R/W heads.

Transfer Time

It is the time taken to transfer the data.

Disk Access Time

Disk access time is given as,

Disk Access Time = Rotational Latency + Seek Time + Transfer Time

Disk Response Time

It is the average of time spent by each request waiting for the IO operation.

Purpose of Disk Scheduling

The main purpose of disk scheduling algorithm is to select a disk request from the queue of IO requests and decide the schedule when this request will be processed.

Goal of Disk Scheduling Algorithm

- Fairness
- High throughout
- Minimal traveling head time

Disk Scheduling Algorithms

The list of various disks scheduling algorithm is given below. Each algorithm is carrying some advantages and disadvantages. The limitation of each algorithm leads to the evolution of a new algorithm.

- FCFS scheduling algorithm
- SSTF (shortest seek time first) algorithm
- SCAN scheduling
- C-SCAN scheduling
- LOOK Scheduling
- C-LOOK scheduling

FCFS Scheduling Algorithm

It is the simplest Disk Scheduling algorithm. It services the IO requests in the order in which they arrive. There is no starvation in this algorithm, every request is serviced.

Disadvantages

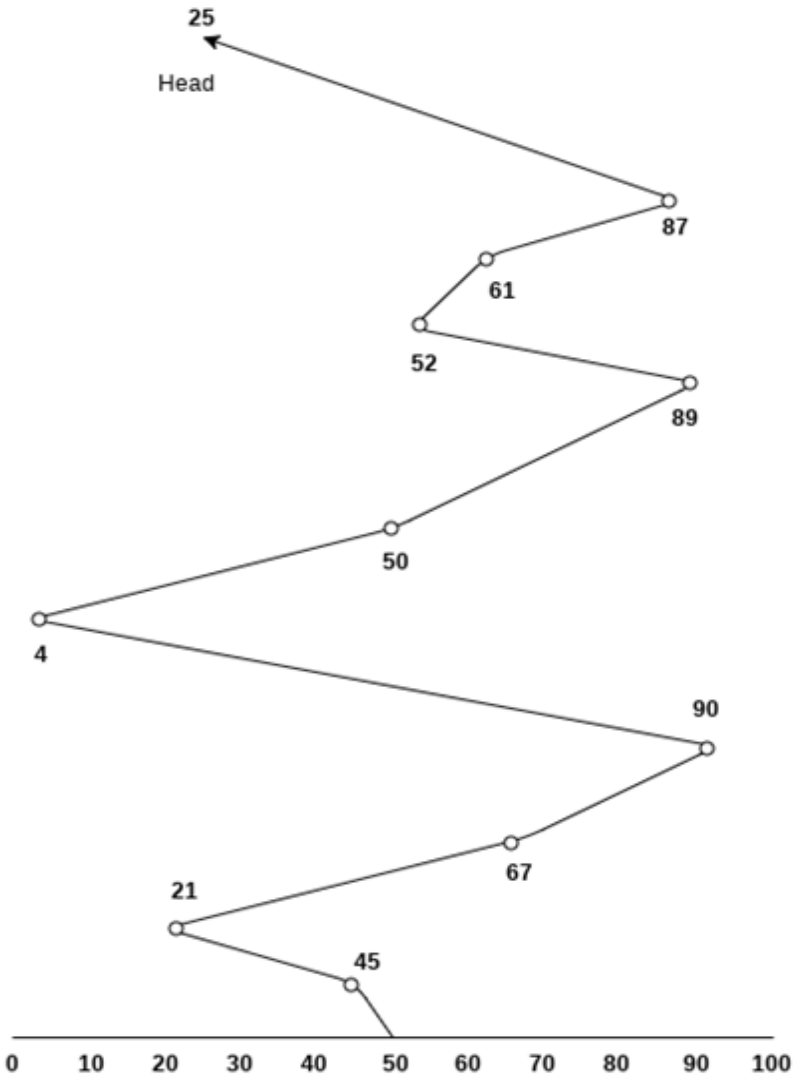
- The scheme does not optimize the seek time.
- The request may come from different processes therefore there is the possibility of inappropriate movement of the head.

Example

Consider the following disk request sequence for a disk with 100 tracks 45, 21, 67, 90, 4, 50, 89, 52, 61, 87, 25

Head pointer starting at 50 and moving in left direction. Find the number of head movements in cylinders using FCFS scheduling.

Solution



= (50-45)+(45-21)+(67-21)+(90-67)+(90-4)+(50-4)+(89-50)+(61-52)+(87-61)+(87-25)

= 5 + 24 + 46 + 23 + 86 + 46 + 49 + 9 + 26 + 62

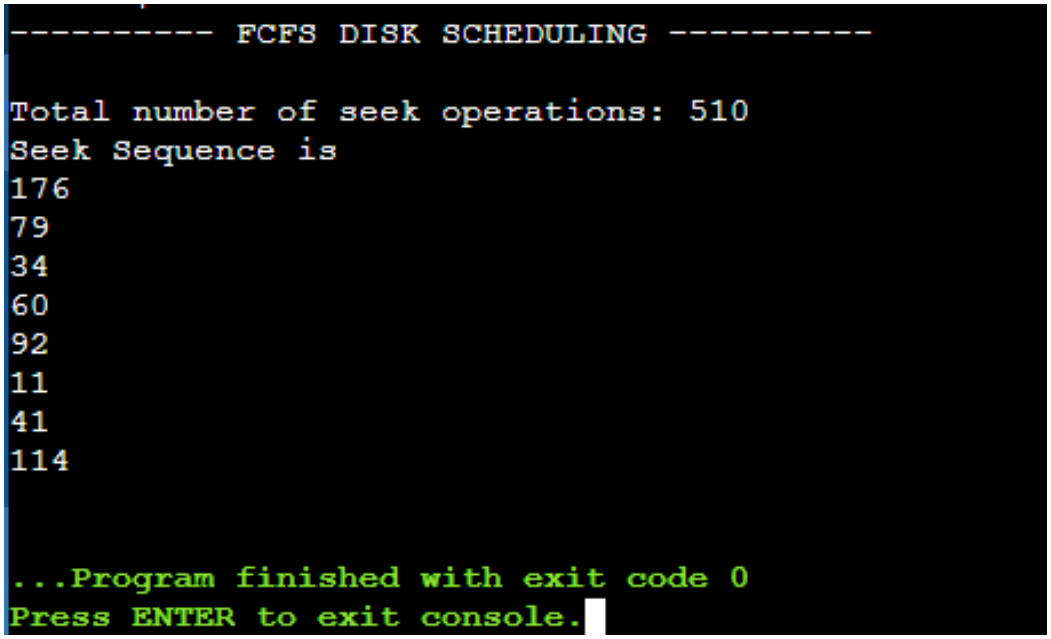
= 376

- **C PROGRAM (FCFS) :**

```
#include <stdio.h>
#include <math.h>

#define SIZE 8

void FCFS(int arr[], int head) {
    int seek_count = 0;
    int cur_track, distance;

    for (int i = 0; i < SIZE; i++) {
        cur_track = arr[i];

        // calculate absolute distance
        distance = abs(head - cur_track);

        // increase the total count
        seek_count += distance;

        // accessed track is now new head
        head = cur_track;
    }

    printf("Total number of seek operations: %d\n", seek_count);

    // Seek sequence would be the same
    // as request array sequence
```

```
    printf("Seek Sequence is\n");

    for (int i = 0; i < SIZE; i++) {
        printf("%d\n", arr[i]);
    }
}

//Driver code
int main() {
    // request array
    int arr[SIZE] = { 176, 79, 34, 60, 92, 11, 41, 114 };
    int head = 50;
    printf("---------- FCFS DISK SCHEDULING ----------\n\n");

    FCFS(arr, head);

    return 0;
}
```

- **OUTPUT:**



- **<u>CONCLUSION:</u> Hence, we have successfully implemented the concept of Disk Scheduling.**