Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

Academic Year : 2023-24

Experiment No 10: Disk scheduling algorithm
Date of Performance:05/04/24
Date of Submission:05/04/24
Name: JENIL KOTADIA
Roll No : 19

AIM: To Study and Implement Disk scheduling algorithm FCFS SSTF

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

THEORY :

FCFS (First-Come, First-Served) Disk Scheduling Algorithm is a simple and easy-to-implement algorithm used in operating systems to manage input/output (I/O) requests from processes to access disk blocks. In this algorithm, the operating system processes the I/O requests in the order in which they arrive in the queue, without any reordering or prioritization.

When a process generates an I/O request, it is added to the end of the queue, and the operating system services the requests in the same order. The requests are serviced one by one until the entire queue is empty.

ADVANTAGES :

1. **Simplicity** - FCFS is a simple algorithm that is easy to implement and understand. It does not require any complex calculations or heuristics.

2. **Fairness** – FCFS provides fairness in processing I/O requests, as it processes them in the order in which they arrive in the queue. This ensures that all requests are processed eventually, and no request is unfairly prioritized over another.

3. **Low overhead** – FCFS has low overhead, as it does not require any additional information or processing. It only requires a simple queue to store and process I/O requests.

DISADVANTAGES:

1. **Poor performance for long requests** – Since FCFS processes requests in the order they arrive, long requests may have to wait for a long time before being processed, leading to higher response times. This can significantly impact the overall performance of the system, especially if there are many long requests in the queue.

2. **Inefficient disk usage** – FCFS does not consider the

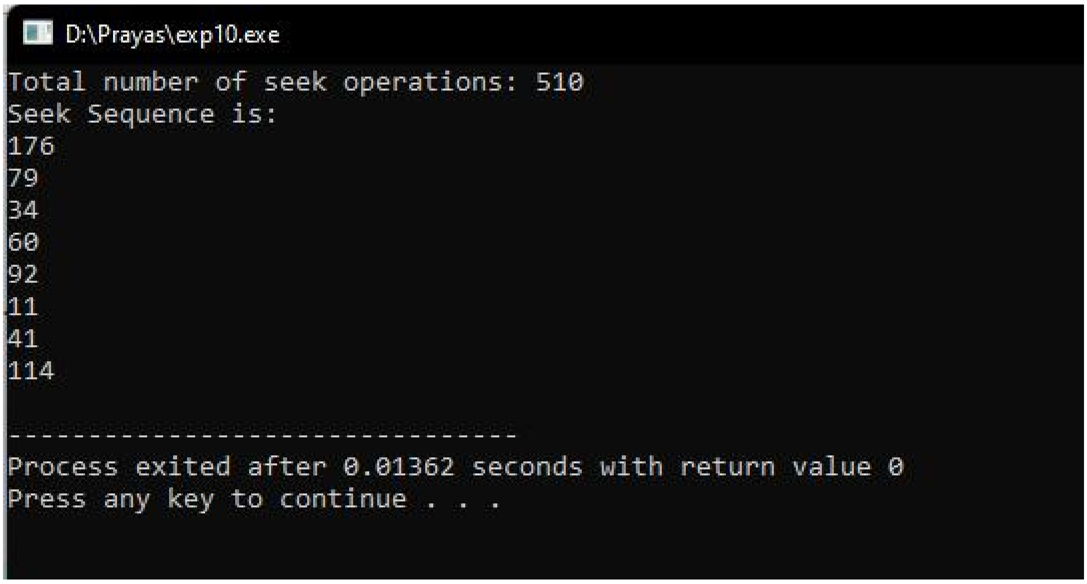
location of the data on the disk, which can lead to inefficient disk usage as requests may not be processed in a way that optimizes disk access. This can result in longer waiting times for requests and increased disk access times.

3. Limited scalability – The algorithm may not scale well in systems with a large number of I/O requests, as the queue can become overwhelmed, and the waiting time for requests can become excessively long. This can lead to decreased system performance and reduced efficiency.

CODE:

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

int 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];
        distance = fabs(head - cur_track);
        seek_count += distance;
        head = cur_track;
    }

    printf("Total number of seek operations: %d\n", seek_count);
    printf("Seek Sequence is:\n");
    for (int i = 0; i < size; i++) {
        printf("%d\n", arr[i]);
    }
}

int main() {
    int arr[8] = {176, 79, 34, 60, 92, 11, 41, 114};
    int head = 50;

    FCFS(arr, head);

    return 0;
}
```

OUTPUT:



```
Total number of seek operations: 510
Seek Sequence is:
176
79
34
60
92
11
41
114

--------------------------------
Process exited after 0.01362 seconds with return value 0
Press any key to continue . . .
```

CONCLUSION :

In conclusion, FCFS disk scheduling algorithm is a simple and fair method for managing input/output requests to disks in operating systems. However, it has several disadvantages, including long waiting times for requests that arrive later and potential starvation of requests that are stuck behind long-running requests.