



Nagar Yuwak Shikshan Sanstha's
Yeshwantrao Chavan College of Engineering
(An Autonomous Institution affiliated to Rashtrasant Tukadoji Maharaj Nagpur University)
Hingna Road, Wanadongri, Nagpur - 441 110
Ph.: 07104-237919, 234623, 329249, 329250 Fax: 07104-232376, Website: www.ycce.edu

Department of Computer Science and Engineering (IOT)

YCCE

Vision

"To become the most preferred institution providing innovative, research and value based, professional education for the society at large".

Mission

YCCE is committed to

- Attract besttalent and create learning ambience
- Practice Innovative teaching-learning & research
- Integrate Industry-Institute Collaborations
- Nurture students towards holistic development and choicest career

Department

Vision of the Department

To be a well-known center for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.

Mission of the Department

To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary arena by developing problem-solving skills through emerging technologies.



Department of Computer Science and Engineering (IOT)

23CT1402		Lab: Operating Systems										
Name of the Student: Parth Bhedurkar		Semester/ Section: 5 A										
Roll No: 52		Enrollment Number: 23071495										

Sr. No.	COs	Course Outcomes	POs											PSOs	
			1	2	3	4	5	6	7	8	9	10	11	PSO 1	PSO2
1	CO1	Demonstrate the ability to execute Linux process management, memory management, and shell commands to manage system resources efficiently.	3	3	3	-		-	-	-	-	-	3	3	
2	CO2	Develop programs utilizing system calls, thread programming, and page replacement algorithms to simulate and analyze operating system functionalities.	3	3	3	-	-	-	-	-	-	-	3	3	
3	CO3	Design and implement process scheduling, memory allocation, and deadlock detection algorithms to address real-world operating system challenges.	3	3	3	-		-	-	-	-	-	3	3	
		Avg	3	3	-		-	-	-	-	-	-	3	3	



Department of Computer Science and Engineering (IOT)

Practical No. 7

Aim: Implement a program to simulate disk scheduling algorithms. A. FCFS B. SSTF

Theory:

Prerequisite: Disk scheduling algorithms.

Given an array of disk track numbers and initial head position, our task is to find the total number of seek operations done to access all the requested tracks if **First Come First Serve (FCFS)** disk scheduling algorithm is used.

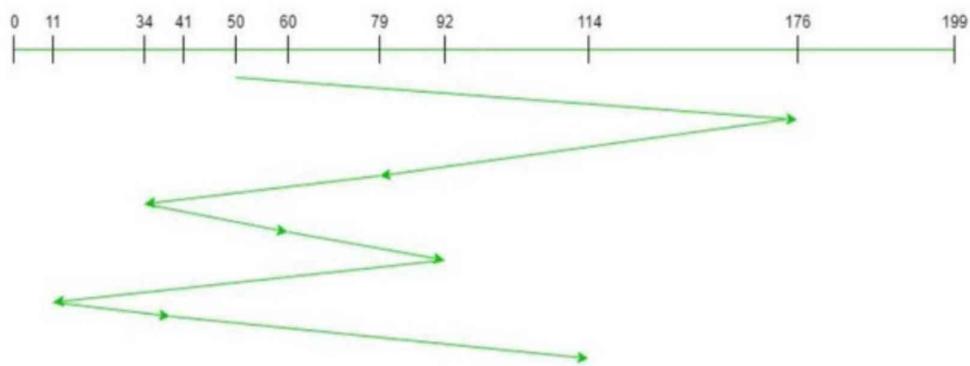
First Come First Serve (FCFS)

FCFS is the simplest disk scheduling algorithm. As the name suggests, this algorithm entertains requests in the order they arrive in the disk queue. The algorithm looks very fair and there is no starvation (all requests are serviced sequentially) but generally, it does not provide the fastest service.

Algorithm:

1. Let Request array represents an array storing indexes of tracks that have been requested in ascending order of their time of arrival. 'head' is the position of disk head.
2. Let us one by one take the tracks in default order and calculate the absolute distance of the track from the head.
3. Increment the total seek count with this distance.
4. Currently serviced track position now becomes the new head position.
5. Go to step 2 until all tracks in request array have not been serviced.

The following chart shows the sequence in which requested tracks are serviced using FCFS.



Code: #include <iostream>

#include <vector>



```
#include <cmath>
#include <algorithm>
#include <climits>
using namespace std;

void FCFS(vector<int> requests, int head) {
    int seek_time = 0;
    cout << "\n--- FCFS Disk Scheduling ---\n";
    cout << "Sequence of Head Movement: " << head;

    for (int i = 0; i < requests.size(); i++) {
        seek_time += abs(requests[i] - head);
        head = requests[i];
        cout << " -> " << head;
    }

    cout << "\nTotal Seek Time: " << seek_time;
    cout << "\nAverage Seek Time: " << (float)seek_time / requests.size() << endl;
}
```



```
void SSTF(vector<int> requests, int head) {  
    int seek_time = 0;  
    int n = requests.size();  
    vector<bool> visited(n, false);  
    cout << "\n--- SSTF Disk Scheduling ---\n";  
    cout << "Sequence of Head Movement: " << head;  
  
    for (int i = 0; i < n; i++) {  
        int min_distance = INT_MAX;  
        int index = -1;  
  
        for (int j = 0; j < n; j++) {  
            if (!visited[j] && abs(requests[j] - head) < min_distance) {  
                min_distance = abs(requests[j] - head);  
                index = j;  
            }  
        }  
  
        visited[index] = true;  
        seek_time += abs(requests[index] - head);  
        head = requests[index];  
    }  
}
```



Department of Computer Science and Engineering (IOT)

```
cout << " -> " << head;  
}  
  
  
cout << "\nTotal Seek Time: " << seek_time;  
cout << "\nAverage Seek Time: " << (float)seek_time / n << endl;  
}  
  
  
// Main Function  
  
int main() {  
    int n, head;  
    cout << "Enter number of disk requests: ";  
    cin >> n;  
  
  
    vector<int> requests(n);  
    cout << "Enter disk request sequence: ";  
    for (int i = 0; i < n; i++)  
        cin >> requests[i];  
  
  
    cout << "Enter initial head position: ";  
    cin >> head;  
  
  
    FCFS(requests, head);
```



Department of Computer Science and Engineering (IOT)

SSTF(requests, head);

return 0;

}



Department of Computer Science and Engineering (IOT)

OUTPUT (SCREEN SHOT) IF ANY:

```
main.cpp
1 #include <iostream>
2 #include <vector>
3 #include <cmath>
4 #include <algorithm>
5 #include <climits>
6 using namespace std;
7
8
9 void FCFS(vector<int> requests, int head) {
10     int seek_time = 0;
11     cout << "\n--- FCFS Disk Scheduling ---\n";
12     cout << "Sequence of Head Movement: " << head;
13
14     for (int i = 0; i < requests.size(); i++) {
15         seek_time += abs(requests[i] - head);
16         head = requests[i];
17         cout << " -> " << head;
18     }
19
20     cout << "\nTotal Seek Time: " << seek_time;
21     cout << "\nAverage Seek Time: " << (float)seek_time / requests.size() << endl;
22 }
23
24
25 void SSTF(vector<int> requests, int head) {
26     int seek_time = 0;
27     int n = requests.size();
28     vector<bool> visited(n, false);
29     cout << "\n--- SSTF Disk Scheduling ---\n";
30     cout << "Sequence of Head Movement: " << head;
31
32     for (int i = 0; i < n; i++) {
33         int min_distance = INT_MAX;
34         int index = -1;
```

```
Enter number of disk requests: 5
Enter disk request sequence: 34 56 78 90 23 45
Enter initial head position:
--- FCFS Disk Scheduling ---
Sequence of Head Movement: 45 -> 34 -> 56 -> 78 -> 90 -> 23
Total Seek Time: 134
Average Seek Time: 26.8

--- SSTF Disk Scheduling ---
Sequence of Head Movement: 45 -> 34 -> 23 -> 56 -> 78 -> 90
Total Seek Time: 89
Average Seek Time: 17.8

...Program finished with exit code 0
Press ENTER to exit console.[]
```



Nagar Yuwak Shikshan Sanstha's
Yeshwantrao Chavan College of Engineering
(An Autonomous Institution affiliated to Rashtrasant Tukadoji Maharaj Nagpur University)
Hingna Road, Wanadongri, Nagpur - 441 110
Ph.: 07104-237919, 234623, 329249, 329250 Fax: 07104-232376, Website: www.ycce.edu
Department of Computer Science and Engineering (IOT)

Conclusion: Simulate the page replacement algorithm is done successfully.