

FAIZAN CHOUDHARY

20BCS021

OS LAB

5th May 2022

CODE: (code pasted in this format for readability)

```
#include <iostream>
#include <algorithm>
#include <math.h>
using namespace std;
// head movement data for FCFS (index 0), SJF (index 1), and Elevator (index 2)
int **head_movement = new int*[3];
int *total_head_movement = new int [3];
void sort_sstf (int *disk, int n, int start_pos) {
    int i, j, temp, min_index;
    for (i=0; i<n-1; i++) {
        min_index = i;
        for (j=i+1; j<n; j++) {
            if (abs (disk[j] - start_pos) < abs (disk[min_index] - start_pos))
                min_index = j;
        }
        temp = disk[i];
        disk[i] = disk[min_index];
        disk[min_index] = temp;
    }
}
void sort_elevator (int *disk, int n, int start_pos) {
    int i, j;
    int left_idx = 0, right_idx = 0;
    // partitioning disk elements into two halves, left and right
    int left[n], right[n];
    for (i=0; i<n; i++) {
        if (disk[i] <= start_pos) {
            left [left_idx++] = disk[i];
        }
        else {
            right [right_idx++] = disk[i];
        }
    }
    // sorting them according to distance from start_pos
    sort (left, left + left_idx, greater<int>());
    sort (right, right + right_idx);
    // merging them back
    for (i=0; i<left_idx; i++)
        disk[i] = left[i];
    for (i=0; i<right_idx; i++)
        disk[i+left_idx] = right[i];
}
```

```

void display_pointer_movement (int *disk, int n, int start_pos) {
    int i;
    cout<<"\nPointer movement: ";
    for (i=0; i<n; i++) {
        if (i == 0)
            cout<<start_pos<<" -> "<<disk[i]<<" -> ";
        else {
            if (i == n-1)
                cout<<disk[i];
            else
                cout<<disk[i]<<" -> ";
        }
    }
    cout<<endl;
}

void FCFS (int *disk, int n, int start_pos) {
    for (int i=0; i<n; i++) {
        head_movement[0][i] = abs (disk[i] - start_pos);
        start_pos = disk[i];
        total_head_movement[0] += head_movement[0][i];
    }
}

void Elevator (int *disk, int n, int start_pos) {
    // sorting data for elevator movement
    sort_elevator (disk, n, start_pos);
    for (int i=0; i<n; i++) {
        head_movement[2][i] = abs (disk[i] - start_pos);
        start_pos = disk[i];
        total_head_movement[2] += head_movement[2][i];
    }
}

void SSTF (int *disk, int n, int start_pos) {
    // sorting data for SSTF
    sort_sstf (disk, n, start_pos);
    for (int i=0; i<n; i++) {
        head_movement[1][i] = abs (disk[i] - start_pos);
        start_pos = disk[i];
        total_head_movement[1] += head_movement[1][i];
    }
}

int main() {
    cout<<"\nFAIZAN CHOUDHARY\n20BCS021\n";
    cout<<"\nFirst Come First Served (FCFS), Shortest Seek Time First (SSTF) and Elevator
Disk Scheduling\n";
    int n, i;
    cout<<"\nEnter the number of disk requests in the queue: ";
    cin>>n;
    for (i=0; i<3; i++) {
        head_movement[i] = new int [n];
        total_head_movement[i] = 0;
    }
    // initializing disk requests for different scheduling algorithms, since they require
    sorting and partitioning
    int *disk_requests = new int [n];

```

```

int *disk_requests_sstf = new int [n];
int *disk_requests_elevator = new int [n];
cout<<"\nEnter the disk requests: ";
for (i=0; i<n; i++) {
    cin>>disk_requests[i];
    disk_requests_sstf[i] = disk_requests[i];
    disk_requests_elevator[i] = disk_requests[i];
}
int start_position;
cout<<"\nEnter the starting position of the disk head: ";
cin>>start_position;
FCFS(disk_requests, n, start_position);
SSTF(disk_requests_sstf, n, start_position);
Elevator(disk_requests_elevator, n, start_position);
for (int i=0; i<3; i++) {
    if (i==0) {
        cout<<"\nFCFS:\n";
        display_pointer_movement (disk_requests, n, start_position);
    }
    else if (i==1) {
        cout<<"\nSSTF:\n";
        display_pointer_movement (disk_requests_sstf, n, start_position);
    }
    else if (i==2) {
        cout<<"\nElevator:\n";
        display_pointer_movement (disk_requests_elevator, n, start_position);
    }
    cout<<"Total head movement: ";
    for (int j=0; j<n; j++) {
        if (j == n-1)
            cout<<head_movement[i][j]<<" = ";
        else
            cout<<head_movement[i][j]<<" + ";
    }
    cout<<total_head_movement[i]<<" tracks"<<endl;
}
cout<<endl;
return 0;
}

```

OUTPUT:

FAIZAN CHOUDHARY
20BCS021

First Come First Served (FCFS), Shortest Seek Time First (SSTF) and Elevator Disk Scheduling

Enter the number of disk requests in the queue: 8

Enter the disk requests: 98 183 37 122 14 124 65 67

Enter the starting position of the disk head: 53

FCFS:

Pointer movement: 53 -> 98 -> 183 -> 37 -> 122 -> 14 -> 124 -> 65 -> 67
Total head movement: $45 + 85 + 146 + 85 + 108 + 110 + 59 + 2 = 640$ tracks

SSTF:

Pointer movement: 53 -> 65 -> 67 -> 37 -> 14 -> 98 -> 122 -> 124 -> 183
Total head movement: $12 + 2 + 30 + 23 + 84 + 24 + 2 + 59 = 236$ tracks

Elevator:

Pointer movement: 53 -> 37 -> 14 -> 65 -> 67 -> 98 -> 122 -> 124 -> 183
Total head movement: $16 + 23 + 51 + 2 + 31 + 24 + 2 + 59 = 208$ tracks

FAIZAN CHOUDHARY
20BCS021

First Come First Served (FCFS), Shortest Seek Time First (SSTF) and Elevator Disk Scheduling

Enter the number of disk requests in the queue: 4

Enter the disk requests: 65 40 18 78

Enter the starting position of the disk head: 30

FCFS:

Pointer movement: 30 -> 65 -> 40 -> 18 -> 78
Total head movement: $35 + 25 + 22 + 60 = 142$ tracks

SSTF:

Pointer movement: 30 -> 40 -> 18 -> 65 -> 78
Total head movement: $10 + 22 + 47 + 13 = 92$ tracks

Elevator:

Pointer movement: 30 -> 18 -> 40 -> 65 -> 78
Total head movement: $12 + 22 + 25 + 13 = 72$ tracks