VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

OPERATING SYSTEMS

LAB 5

AIM: **To learn about disk scheduling algorithms and implement them in simple scenarios.**

20BDS0405 (BIMAL PARAJULI)

FCFS:

Write a program to find the total seek distance using First Come First Serve based Disk Scheduling algorithm

Input:

Head start:

53

Number of Requests:

8

Request Queue:

98

183

37

122

14

124

65

67

Output:

640

Code:

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

void gethead();   // Function to get the starting position of
the read/write head from the user.
void getnumber(); // Function to get the number of requests
from the user.
void getdata();   // Function to get the array of actual
requests from the user.
void fcfs();      // Function to calculate and display the
total seek time using FCFS algorithm.

int n;                // n           = number of requests
coming to the disk.
int head;             // head        = The starting position
of the reader head of the disk
int upper_limit = 199; // upper_limit = Upper limit of number
of sectors in disk.
int lower_limit = 0;   // lower_limit = Lower Limit of number
of sectors in disk.
int *request_array;    // Array containing the requests in
order.

// Main function starts here
int main()
{
    gethead();
    getnumber();
    getdata();
    fcfs();

    return 0;
}

// Definition of above declared functions:
```

```c
void gethead()
{
    scanf(" %d", &head);
}

void getnumber()
{
    scanf(" %d", &n);
}

void getdata()
{
    request_array = (int *)malloc((n + 1) * sizeof(int));

    for (int i = 0; i < n; i++)
    {
        scanf(" %d", &request_array[i]);
    }
}

void fcfs()
{
    int temp, total_seek, head_position, past_position;
    head_position = head;

    total_seek = 0;
    past_position = head_position;
    for (int i = 0; i < n; i++)
    {
        total_seek += abs(request_array[i] - past_position);
        past_position = request_array[i];
    }
    printf("%d", total_seek);

    return 0;
}
```

[redacted]

SSTF:

Write a program to find the total seek distance using Shortest Seek Time First based Disk Scheduling algorithm

Input:

Head start:

53

Number of Requests:

8

Request Queue:

98

183

37

122

14

124

65

67

Output:

236

Code:

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

```c
void gethead();   // Function to get the starting position of
the read/write head from the user.
void getnumber(); // Function to get the number of requests
from the user.
void getdata();   // Function to get the array of actual
requests from the user.
void sstf();      // Function to calculate and display the
total seek time using SSTF algorithm.

int n;                // n           = number of requests
coming to the disk.
int head;             // head        = The starting position
of the reader head of the disk
int upper_limit = 199; // upper_limit = Upper limit of number
of sectors in disk.
int lower_limit = 0;   // lower_limit = Lower Limit of number
of sectors in disk.
int *request_array;    // Array containing the requests in
order.

// Main function starts here
int main()
{
    gethead();
    getnumber();
    getdata();
    scan();

    return 0;
}

// Definition of above declared functions:
void gethead()
{
    scanf(" %d", &head);
```

```
}

void getnumber()
{
    scanf(" %d", &n);
}

void getdata()
{
    request_array = (int *)malloc((n + 1) * sizeof(int));

    for (int i = 0; i < n; i++)
    {
        scanf(" %d", &request_array[i]);
    }
}

void sstf()
{
    int temp, lowest, highest, head_position, total;
    head_position = head;

    total = NULL;
    printf("%d", total);
}
```

SCAN:

Write a program to find the total seek distance using SCAN Disk Scheduling algorithm

Input:

Head start:

53

Number of Requests:

8

Request Queue:

98

183

37

122

14

124

65

67

Output:

236

Code:

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

```c
void gethead();   // Function to get the starting position of
the read/write head from the user.
void getnumber(); // Function to get the number of requests
from the user.
void getdata();   // Function to get the array of actual
requests from the user.
void scan();      // Function to calculate and display the
total seek time using SCAN algorithm.

int n;                // n           = number of requests
coming to the disk.
int head;             // head        = The starting position
of the reader head of the disk
int upper_limit = 199; // upper_limit = Upper limit of number
of sectors in disk.
int lower_limit = 0;   // lower_limit = Lower Limit of number
of sectors in disk.
int *request_array;    // Array containing the requests in
order.

// Main function starts here
int main()
{
    gethead();
    getnumber();
    getdata();
    scan();

    return 0;
}

// Definition of above declared functions:
void gethead()
{
    scanf(" %d", &head);
}
```

```
void getnumber()
{
    scanf(" %d", &n);
}

void getdata()
{
    request_array = (int *)malloc((n + 1) * sizeof(int));

    for (int i = 0; i < n; i++)
    {
        scanf(" %d", &request_array[i]);
    }
}

void scan()
{
    int temp, lowest, highest, head_position, total;

    head_position = head;

    lowest = request_array[0];
    highest = request_array[n - 1];

    for (int i = 0; i < n; i++)
    {
        if (request_array[i] > highest)
        {
            highest = request_array[i];
        }
        if (request_array[i] < highest)
        {
            lowest = request_array[i];
        }
    }
```

```
    total = (highest - 0) + (head_position - 0);
    printf("%d", total);
}
```