

**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:

#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:

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;

}

## 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>

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>

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);

}