

EXPERIMENT-39

Develop a C program to simulate C-SCAN disk scheduling algorithm.

AIM:-

The aim of this program is to simulate the **C-SCAN (Circular SCAN)** disk scheduling algorithm. In C-SCAN, the disk arm moves in one direction servicing requests until it reaches the end of the disk, then jumps to the beginning of the disk and services requests in the same direction.

ALGORITHM:-

1. **Input:** A list of disk requests, the initial position of the disk head, and the direction of movement (left or right).
2. **Sort the Requests:** Sort the disk requests in ascending order.
3. **Service the Requests:** Move the disk arm towards one end, servicing requests as it goes. When it reaches the end, jump to the start of the disk and continue servicing the remaining requests.
4. **Calculate the Total Seek Time:** Add up all the individual seek times.
5. **Display the Disk Head Movement:** Display the movement of the disk arm from one request to the next, and the total seek time.

PROCEDURE:-

1. Accept the disk head position and the disk requests.
2. Sort the disk requests in ascending order.
3. Move the disk arm in the specified direction and service the requests.
4. Once the disk arm reaches the end, move it back to the start (without servicing) and continue servicing the remaining requests in the same direction.
5. Display the disk arm's movement and calculate the total seek time.

CODE:-

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

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

```
// Function to sort an array
```

```
void sort(int arr[], int n) {
```

```
    for (int i = 0; i < n-1; i++) {
```

```
        for (int j = i+1; j < n; j++) {
```

```
            if (arr[i] > arr[j]) {
```

```
                int temp = arr[i];
```

```
                arr[i] = arr[j];
```

```
                arr[j] = temp;
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

```
// Function to simulate C-SCAN Disk Scheduling algorithm
```

```
void C_SCAN(int requests[], int n, int initialPosition, int diskSize) {
```

```
    int totalSeekTime = 0;
```

```
    int currentPosition = initialPosition;
```

```
// Sort the requests in ascending order
```

```

sort(requests, n);

// Calculate seek time

int seekTime = 0;

// Move right to the end first

for (int i = 0; i < n; i++) {

    if (requests[i] >= currentPosition) {

        seekTime += abs(currentPosition - requests[i]);

        currentPosition = requests[i];

    }

}

// Move to the last track

seekTime += abs(currentPosition - (diskSize - 1));

currentPosition = diskSize - 1;


// Jump to the first track (simulating the circular move)

seekTime += abs(currentPosition - 0);

currentPosition = 0;


// Move right from the beginning now

for (int i = 0; i < n; i++) {

    if (requests[i] > currentPosition) {

        seekTime += abs(currentPosition - requests[i]);

        currentPosition = requests[i];

    }

}

```

```

    }

}

// Display disk head movements and total seek time

printf("Total Seek Time: %d\n", seekTime);

}

int main() {

    int n, initialPosition, diskSize;

    // Accept the number of disk requests and disk size

    printf("Enter the number of disk requests: ");

    scanf("%d", &n);

    int requests[n];

    printf("Enter the disk requests: ");

    for (int i = 0; i < n; i++) {

        scanf("%d", &requests[i]);

    }

    // Accept the initial position of the disk head

    printf("Enter the initial position of the disk head: ");

    scanf("%d", &initialPosition);

    // Accept the disk size

```

```

printf("Enter the size of the disk: ");

scanf("%d", &diskSize);

// Simulate C-SCAN disk scheduling

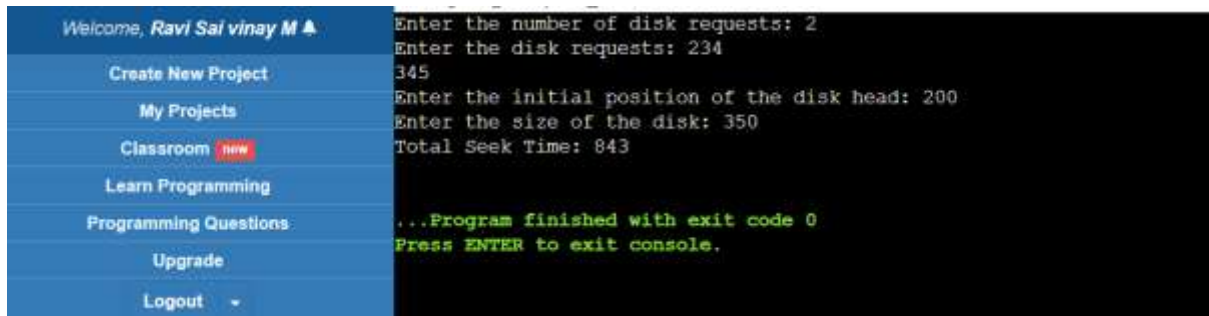
C_SCAN(requests, n, initialPosition, diskSize);

return 0;

}

```

OUTPUT:-



The screenshot shows a web application on the left and a terminal window on the right. The web application has a blue sidebar with the following menu items: 'Welcome, Ravi Sai vinay M ▲', 'Create New Project', 'My Projects', 'Classroom new', 'Learn Programming', 'Programming Questions', 'Upgrade', and 'Logout ▾'. The terminal window on the right displays the following text:

```

Enter the number of disk requests: 2
Enter the disk requests: 234
345
Enter the initial position of the disk head: 200
Enter the size of the disk: 350
Total Seek Time: 843

...Program finished with exit code 0
Press ENTER to exit console.

```

RESULT:-

The program successfully simulates the **C-SCAN disk scheduling algorithm**. It processes disk requests by moving the disk arm in one direction, servicing requests until the end of the disk, and then jumps back to the beginning to continue servicing the remaining requests.