Department Of Computer Science and Engineering

Course Title: Operating System Lab

Course Code: CSE 406

Title: CPU Scheduling Disk Sheduling Algorithm

Submitted To   Submitted By

Atia Rahman Orthi   Md. Atikul Islam Atik
Lecturer   Reg No:21201063
Department Of Computer Science &   Sec:B1
Engineering

Submission Date: 7.3.2025

INPUT:

Req Seq= {176,79,34,60,92,11,41,114}

Head:50;

Output:

510

Source Code:

```python
# Disk Scheduling
# First Come First Serve (FCFS) Scheduling

def FCFS(sequences, head):
    seek_count = 0
    current_head = head
    current_sequence =[head]

    for seq in sequences:
        distance = abs(current_head - seq)
        seek_count += distance
        current_head = seq
        current_sequence.append(seq)

    return seek_count, current_sequence


def take_input():
    head = int(input("Enter the initial head position: "))
    n = int(input("Enter the number of seqence: "))
    req_sequences = []

    for i in range(n):
        req = int(input(f'Enter {i+1} sequence number: '))
        req_sequences.append(req)
    return req_sequences, head

def print_sequence(sequences):
    for seq in sequences:
        print(seq, end=" ---> ")

if __name__ == "__main__":
    inputs = take_input()
    req_sequences = inputs[0]
    head = inputs[1]

    res = FCFS(req_sequences, head)
    print('Total Seek count = ', res[0])
    print_sequence(res[1])
```

[Live Link of Code](#)

Output :


```
 atik   os/lab4   main !?    10:34 PM
 python -u "/home/atik/Codes/python/os/lab4/disk-sheduling/fcfs-disk-sheduling.py"
Enter the initial head position: 50
Enter the number of seqence: 8
Enter 1 sequence number: 176
Enter 2 sequence number: 79
Enter 3 sequence number: 34
Enter 4 sequence number: 60
Enter 5 sequence number: 92
Enter 6 sequence number: 11
Enter 7 sequence number: 41
Enter 8 sequence number: 114
Total Seek count =  510
50 ---> 176 ---> 79 ---> 34 ---> 60 ---> 92 ---> 11 ---> 41 ---> 114 --->

 atik   os/lab4   main !?    10:34 PM
```

Algorithm:

Input the Array: Take an array of values as input.
Input the Head Value: Initialize the head with a given starting value.
Process the Array:

- Iterate through each value in the array.
- In each iteration:
    - Calculate the difference between the current value and the previous head value.
    - Update head to the current value.

Output the Result: Display the computed differences for each step.

Conclusion:

Disk scheduling algorithms manage requests for disk input and output, and are important for efficient system performance.