

Fr. Conceicao Rodrigues College of Engineering Department of Computer Engineering			
Student's Roll No	9913	Students Name	Mark Lopes
Date of Performance		SE Computer – Div	A

Aim: Study Disk Management

Lab Outcome:

CSL403.6: Implement various Disk Management techniques and evaluate their performance.

Problem Statements:

Implement Disk Management Algorithms

(a)FCFS (b)SSTF (c)SCAN

Given the current head position and future disk block references wrt tracks or cylinders. Calculate the seek length based on above algorithms. Show the sequence in which the disk blocks will be accessed and no of tracks traversed in each algorithm.

References:

<https://www.geeksforgeeks.org/disk-scheduling-algorithms/>
https://www.youtube.com/watch?v=9uoa_p8q47Y

```
#First come first serve

def FCFS(requests, initial_head):

    total_head_movement = 0

    current_head = initial_head

    print(current_head, end=" > ")

    for request in requests:

        total_head_movement += abs(request - current_head)

        current_head = request

        print(current_head, end=" > ")

    print("\nThe total head movement is:", total_head_movement)
```

```
#Shortest Seek Time First

def SSTF(requests, initial_head):

    total_head_movement = 0

    current_head = initial_head

    print(current_head, end=" > ")

    while requests:

        nearest = min(requests, key=lambda x: abs(x - current_head))

        total_head_movement += abs(nearest - current_head)

        current_head = nearest

        print(current_head, end=" > ")

        requests.remove(nearest)

    print("\nThe total head movement is:", total_head_movement)


# SCAN Disk Scheduling

def SCAN(requests, initial_head, prev_head, block_size):

    total_head_movement = 0

    current_head = initial_head

    print(current_head, end=" > ")

    # Divide requests into lesser and greater parts

    lesser_arr = [req for req in requests if req < current_head]

    greater_arr = [req for req in requests if req >= current_head]

    # Add boundary blocks to the lists

    lesser_arr.append(block_size["start"])

    greater_arr.append(block_size["end"])
```

```
# Sort the lists

lesser_arr.sort(reverse=True)

greater_arr.sort()

if initial_head > prev_head:

    lesser_arr.pop() # Remove the start boundary block

    for ele in greater_arr:

        total_head_movement += abs(ele - current_head)

        current_head = ele

        print(current_head, end=" > ")

    for ele in lesser_arr:

        total_head_movement += abs(ele - current_head)

        current_head = ele

        print(current_head, end=" > ")

else:

    greater_arr.pop() # Remove the end boundary block

    for ele in lesser_arr:

        total_head_movement += abs(ele - current_head)

        current_head = ele

        print(current_head, end=" > ")

    for ele in greater_arr:

        total_head_movement += abs(ele - current_head)

        current_head = ele

        print(current_head, end=" > ")
```

```
print("\nThe total head movement is:", total_head_movement)

# Test the disk scheduling algorithms

requests = [98, 183, 37, 122, 14, 124, 65, 67]

initial_head = 53

prev_head = 24

block_size = {"start": 0, "end": 200}

print("FCFS:")

FCFS(requests.copy(), initial_head)

print("\nSSTF:")

SSTF(requests.copy(), initial_head)

print("\nSCAN:")

SCAN(requests.copy(), initial_head, prev_head, block_size)
```

```
PS C:\Users\Mark Lopes\Desktop\college\Sem_4\Os\lab9> py
FCFS:
53 > 98 > 183 > 37 > 122 > 14 > 124 > 65 > 67 >
The total head movement is: 640

SSTF:
53 > 65 > 67 > 37 > 14 > 98 > 122 > 124 > 183 >
The total head movement is: 236

SCAN:
53 > 65 > 67 > 98 > 122 > 124 > 183 > 200 > 37 > 14 >
The total head movement is: 333
PS C:\Users\Mark Lopes\Desktop\college\Sem_4\Os\lab9>
```

On time Submission(2)	Knowledge of Topic(4)	Implementation and Demonstraion(4)	Total (10)
Signature of Faculty		Date of Submission	