**Batch: A-3 Roll No.: 16010122104**

**Experiment No. 09**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

|  |
| --- |
| **TITLE: Disk Scheduling Algorithms** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**AIM:** Implementation of Disk Scheduling Algorithm like FCFS, SSTF, SCAN, CSCAN, LOOK.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected Outcome of Experiment:**

**CO 4.** To understand various Memory, I/O and File management techniques.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. **Silberschatz A., Galvin P., Gagne G. “Operating Systems Principles”, Willey Eight edition.**
2. **Achyut S. Godbole , Atul Kahate “Operating Systems” McGraw Hill Third**

**Edition.**

1. **William Stallings, “Operating System Internal & Design Principles”, Pearson.**
2. **Andrew S. Tanenbaum, “Modern Operating System”, Prentice Hall.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Pre Lab/ Prior Concepts:**

* Knowledge of disk scheduling algorithm.
* Calculation of seek time and transfer time etc.

**Description of the application to be implemented**:

FCFS (First-Come, First-Served):

Requests are processed in the order they arrive.

Simple and straightforward but can lead to long wait times (convoy effect).

LOOK Scheduling:

The disk arm moves towards the end of the disk in one direction, servicing requests until there are no more requests in that direction, and then it reverses direction.

More efficient than SCAN as it only goes to the requested cylinders.

**Implementation details:** (printout of code)

class DiskScheduler:

    def \_\_init\_\_(self, requests, start\_position):

        self.requests = sorted(requests)  # Sort requests for LOOK

        self.start\_position = start\_position

        self.head\_movement = 0

    def fcfs(self):

        current\_position = self.start\_position

        total\_head\_movement = 0

        print("FCFS Scheduling:")

        for request in self.requests:

            total\_head\_movement += abs(request - current\_position)

            print(f"Moving from {current\_position} to {request}")

            current\_position = request

        print(f"Total head movement: {total\_head\_movement}")

    def look(self):

        current\_position = self.start\_position

        total\_head\_movement = 0

        left\_requests = []

        right\_requests = []

        # Divide requests into two categories

        for request in self.requests:

            if request < current\_position:

                left\_requests.append(request)

            else:

                right\_requests.append(request)

        # Process right requests

        for request in right\_requests:

            total\_head\_movement += abs(request - current\_position)

            print(f"Moving from {current\_position} to {request}")

            current\_position = request

        # Process left requests

        if left\_requests:

            total\_head\_movement += abs(current\_position - left\_requests[-1])

            current\_position = left\_requests[-1]

            print(f"Moving from {current\_position + (1 if right\_requests else 0)} to {current\_position}")

            for request in reversed(left\_requests):

                total\_head\_movement += abs(request - current\_position)

                print(f"Moving from {current\_position} to {request}")

                current\_position = request

        print(f"Total head movement: {total\_head\_movement}")

# Example Usage

if \_\_name\_\_ == "\_\_main\_\_":

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

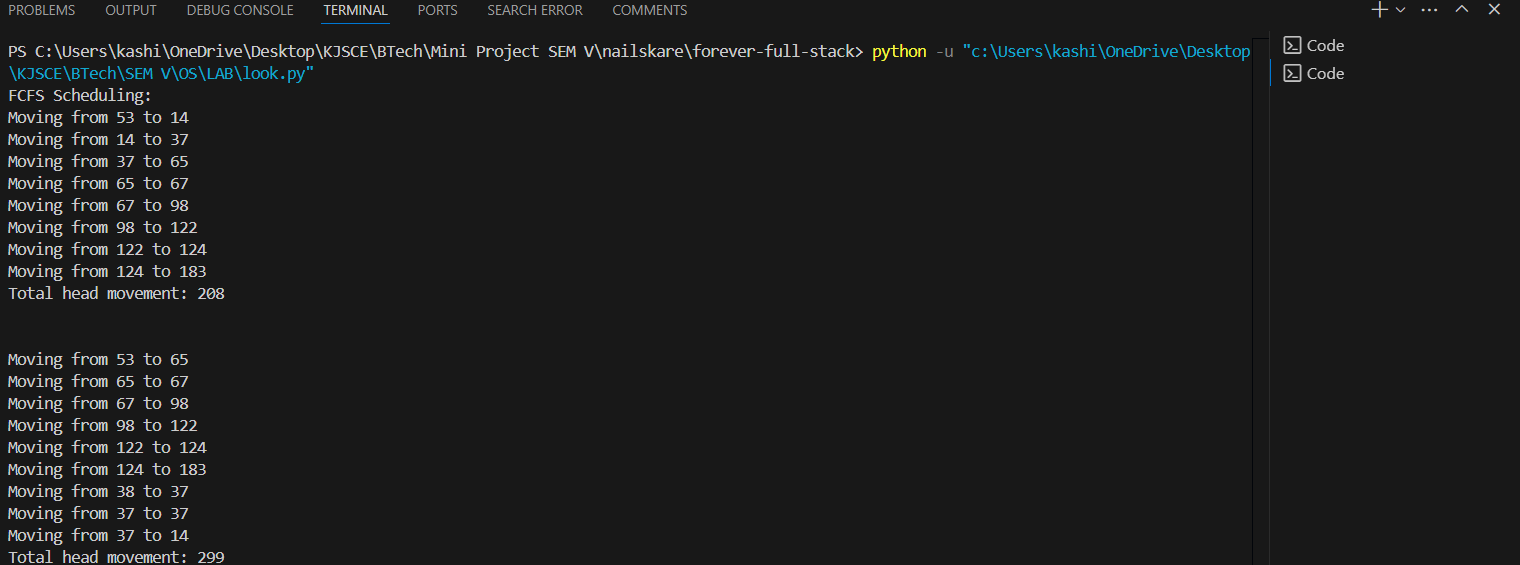
    start\_position = 53

    disk\_scheduler = DiskScheduler(requests, start\_position)

    disk\_scheduler.fcfs()

    print("\n")

    disk\_scheduler.look()



**Conclusion**:

**Post Lab Descriptive Questions**

1. A disk drive has 200 cylinders numbered from 0 to 199. The disk head is initially at cylinder 53. The queue of pending requests in FIFO order is :

98, 183, 37, 122, 14, 124, 65, 67.

Starting from the current head position, what is the total distance travelled (in cylinders) by disk arm to satisfy the requests using CSCAN and Look. Illustrate with figures in each case.

**Ans:**

**Total Distance for CSCAN: 382 cylinders**

**Total Distance for LOOK: 299 cylinders**

**Post Lab Objective Questions**

1. In a hard disk, what rotates about a central spindle
   1. Disk
   2. Platter
   3. Sector
   4. None of the above

**Ans: Platter**

1. The time required to move the disk arm to the required track is known as
   1. Latency time
   2. Access time
   3. Seek time
   4. None of the above

**Ans: Seek Time**