Name:Pranav Patil

Prn:123B2D077

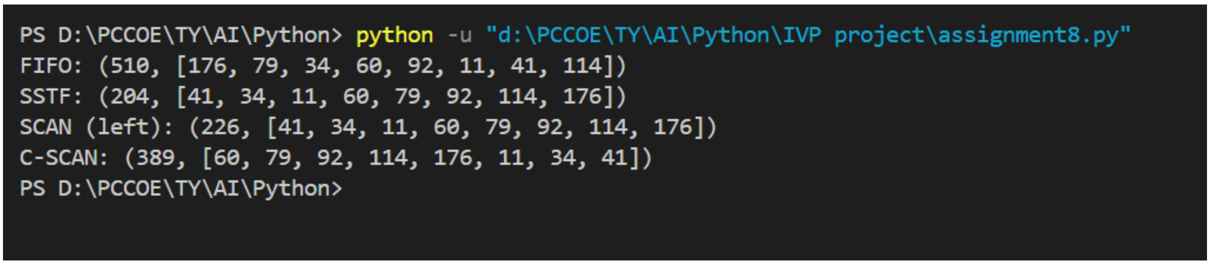
Assignment 8

Write a program to implement disk scheduling algorithms FIFO, SSTF, SCAN, C-SCAN

Source Code:

```
def fifo(requests, head):
    seek_sequence = requests.copy()
    seek_time = 0

    for track in requests:
        seek_time += abs(track - head)
        head = track

    return seek_time, seek_sequence



def sstf(requests, head):
    requests = requests.copy()
    seek_sequence = []
    seek_time = 0

    while requests:
        closest_track = min(requests, key=lambda x: abs(x - head))
        seek_time += abs(closest_track - head)
        head = closest_track
        seek_sequence.append(closest_track)
        requests.remove(closest_track)
    return seek_time, seek_sequence


def scan(requests, head, disk_size, direction="left"):
    requests = sorted(requests)
    seek_sequence = []
    seek_time = 0

    left = [track for track in requests if track < head]
    right = [track for track in requests if track >= head]

    if direction == "left":
        for track in reversed(left):
            seek_time += abs(head - track)
            head = track
            seek_sequence.append(track)

        seek_time += abs(head - 0)
        head = 0

        for track in right:
            seek_time += abs(head - track)
            head = track
            seek_sequence.append(track)

    elif direction == "right":
        for track in right:
            seek_time += abs(head - track)

            head = track

            seek_sequence.append(track)


        seek_time += abs(head - disk_size - 1)

        head = disk_size - 1


        for track in reversed(left):

            seek_time += abs(head - track)

            head = track

            seek_sequence.append(track)


    return seek_time, seek_sequence



def c_scan(requests, head, disk_size):

    requests = sorted(requests)

    seek_sequence = []

    seek_time = 0


    left = [track for track in requests if track < head]

    right = [track for track in requests if track >= head]


    for track in right:

        seek_time += abs(head - track)

        head = track

        seek_sequence.append(track)
    seek_time += abs(head - (disk_size - 1))
    seek_time += disk_size - 1
    head = 0

    for track in left:
        seek_time += abs(head - track)
        head = track
        seek_sequence.append(track)

    return seek_time, seek_sequence


if __name__ == "__main__":
    requests = [176, 79, 34, 60, 92, 11, 41, 114]
    head = 50
    disk_size = 200

    print("FIFO:", fifo(requests, head))
    print("SSTF:", sstf(requests, head))
    print("SCAN (left):", scan(requests, head, disk_size, "left"))
    print("C-SCAN:", c_scan(requests, head, disk_size))
```

Output:



```
PS D:\PCCOE\TY\AI\Python> python -u "d:\PCCOE\TY\AI\Python\IVP project\assignment8.py"
FIFO: (510, [176, 79, 34, 60, 92, 11, 41, 114])
SSTF: (204, [41, 34, 11, 60, 79, 92, 114, 176])
SCAN (left): (226, [41, 34, 11, 60, 79, 92, 114, 176])
C-SCAN: (389, [60, 79, 92, 114, 176, 11, 34, 41])
PS D:\PCCOE\TY\AI\Python>
```