Name : Anto Christopher

Roll no. : 06

TE COMPS

Extra Experiment

LOOK Disk Scheduling

Aim : To implement disk scheduling and calculate average seek time.

Theory:

LOOK is a disk scheduling algorithm used to determine the order in which new disk read and write requests are processed.

The LOOK algorithm is the same as the SCAN algorithm in that it also honors requests on both sweep direction of the disk head, however, this algorithm "Looks" ahead to see if there are any requests pending in the direction of head movement. If no requests are pending in the direction of head movement, then the disk head traversal will be reversed to the opposite direction and requests on the other direction can be served. In LOOK scheduling, the arm goes only as far as final requests in each direction and then reverses direction without going all the way to the end. Consider an example, Given a disk with 200 cylinders (0-199), suppose we have 8 pending requests: 98, 183, 37, 122, 14, 124, 65, 67 and that the read/write head is currently at cylinder 53. In order to complete these requests, the arm will move in the increasing order first and then will move in decreasing order after reaching the end. So, the order in which it will execute is 65, 67, 98, 122, 124, 183, 37, 14.[1]

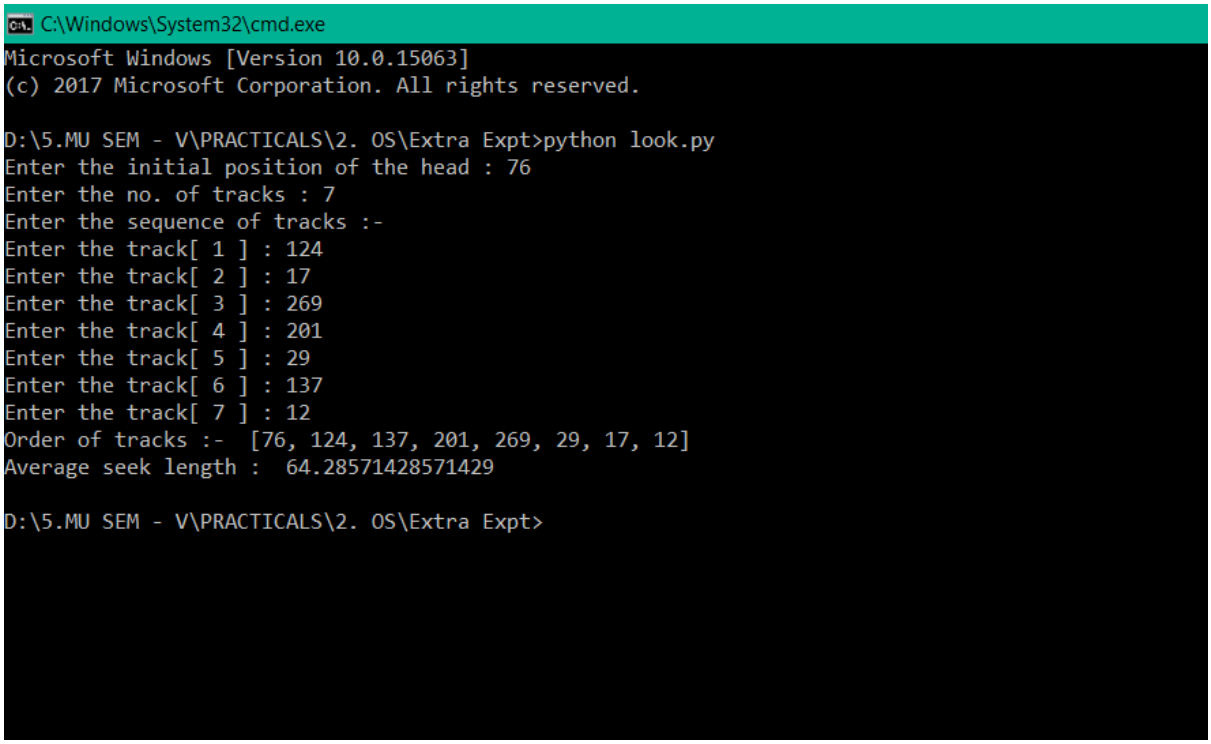
LOOK behaves almost identically to Shortest seek time first (SSTF), but avoids the starvation problem of SSTF. This is because LOOK is biased against the area recently traversed, and heavily favors tracks clustered at the outermost and innermost edges of the platter. LOOK is also biased towards more recently arriving jobs (on average).

Program:

```
def LOOK_alg(init,tmax,tmin,tracks):
      l = len(tracks)
      final_track_list = [init]
      diff_list= []
      recent = init
      for i in range(recent,tmax+1):
            if i in tracks:
                  final_track_list.append(i)
                  diff_list.append(abs(recent-i))
                  recent=i
      i = init-1
      while(i>=tmin):
            if i in tracks:
                  final_track_list.append(i)
                  diff_list.append(abs(recent-i))
                  recent=i
            i=i-1
      return([diff_list,final_track_list])

print("Enter the initial position of the head : ",end="")
init = int(input())
print("Enter the no. of tracks : ",end="")
no_t = int(input())
tracks = []
print("Enter the sequence of tracks :- ")
for i in range (no_t):
      print("Enter the track[",i+1,"] : ",end="")
      inp = i=int(input())
      tracks.append(inp)
dup_tracks = list(tracks)
tmax = max(tracks)
tmin = min(tracks)
res = LOOK_alg(init,tmax,tmin,dup_tracks)
diff_list = res[0]
final_track_list = res[1]
print("Order of tracks :- ",final_track_list)
print("Average seek length : ",(sum(diff_list)/no_t))
```

Output:



Conclusion:

In this experiment we have successfully executed the LOOK disk scheduling algorithm and calculated the average seek time.