Assignment No. 8

Date:

TITLE: Implement the C program for Disk Scheduling Algorithms: SSTF, SCAN, C-Look considering the initial head position moving away from the spindle.

OBJECTIVE:

- Study the concepts of disk scheduling in Linux.
- Learn to implement disk scheduling algorithms in the Linux.

SOFTWARE REQUIREMENTS:

1. Ubuntu 16.04
2. GNU C Compiler

THEORY:

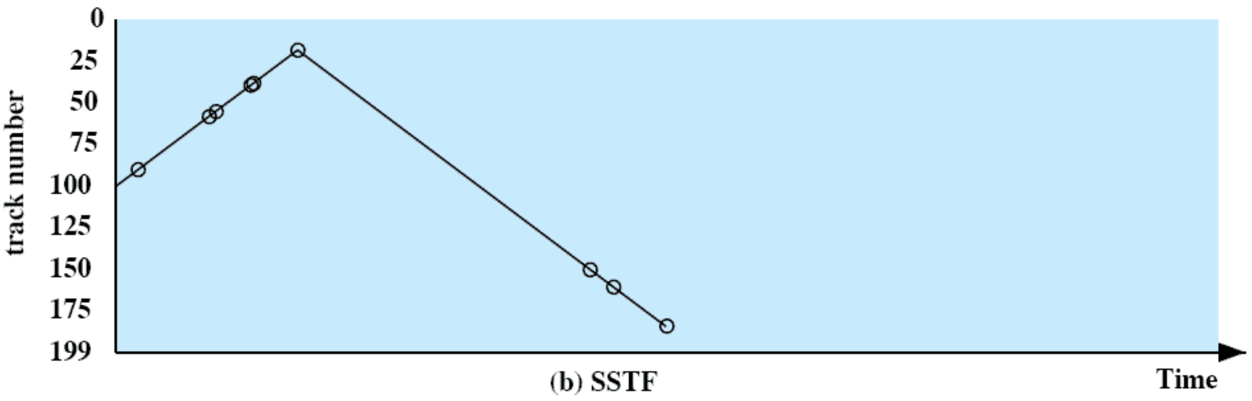
Shortest Service Time First (SSTF)

Select the disk I/O request that requires the least movement of the disk arm from its current position. Always choose the minimum seek time.

Selects the request with the minimum seek time from the current head position. SSTF scheduling is a form of SJF scheduling; may cause starvation of some requests

Example:

To compare various schemes, consider a disk head is initially located at track 100. Assume a disk with 200 tracks and that the disk request queue has random requests in it. The requested tracks, in the order received by the disk scheduler, are 55, 58, 39, 18, 90, 160, 150, 38, 184.



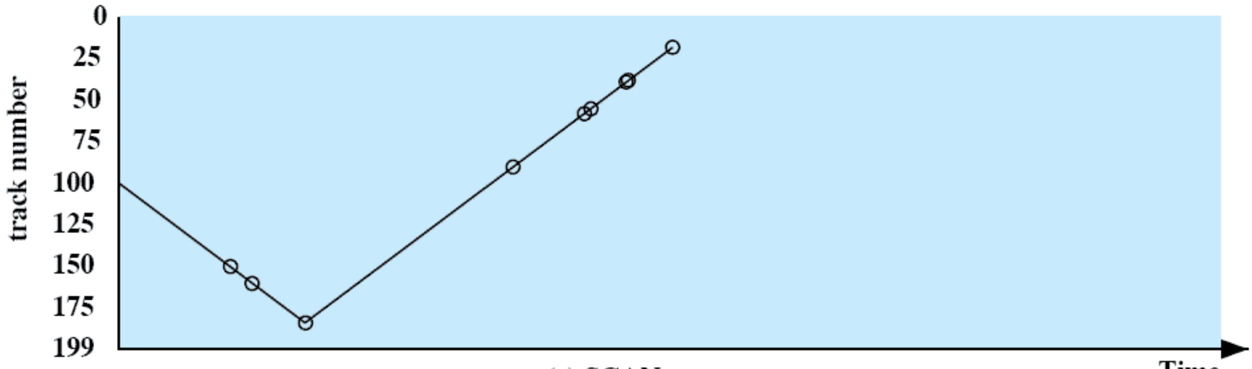
(b) SSTF

SCAN

Arm moves in one direction only, satisfying all outstanding requests until it reaches the last track in that direction then the direction is reversed.

The disk arm starts at one end of the disk, and moves toward the other end, servicing requests until it gets to the other end of the disk, where the head movement is reversed and servicing continues.

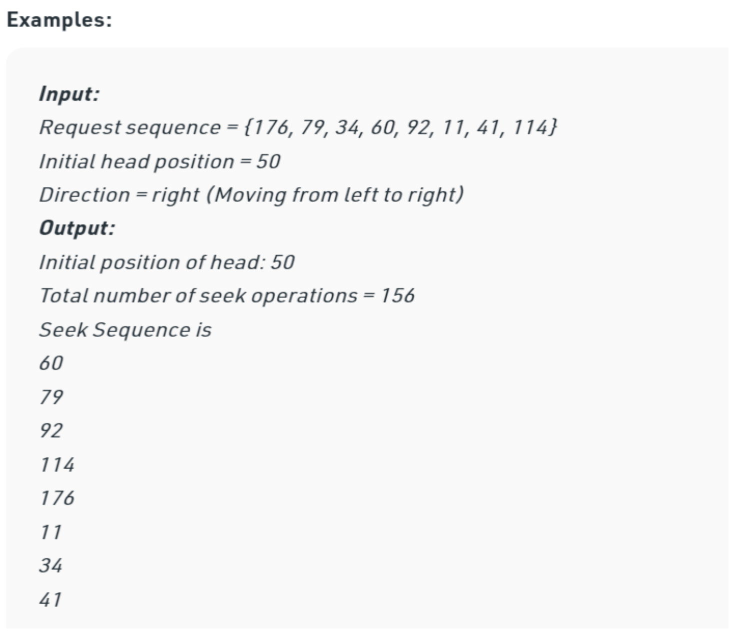
SCAN algorithm Sometimes called the **elevator algorithm**



(c) SCAN

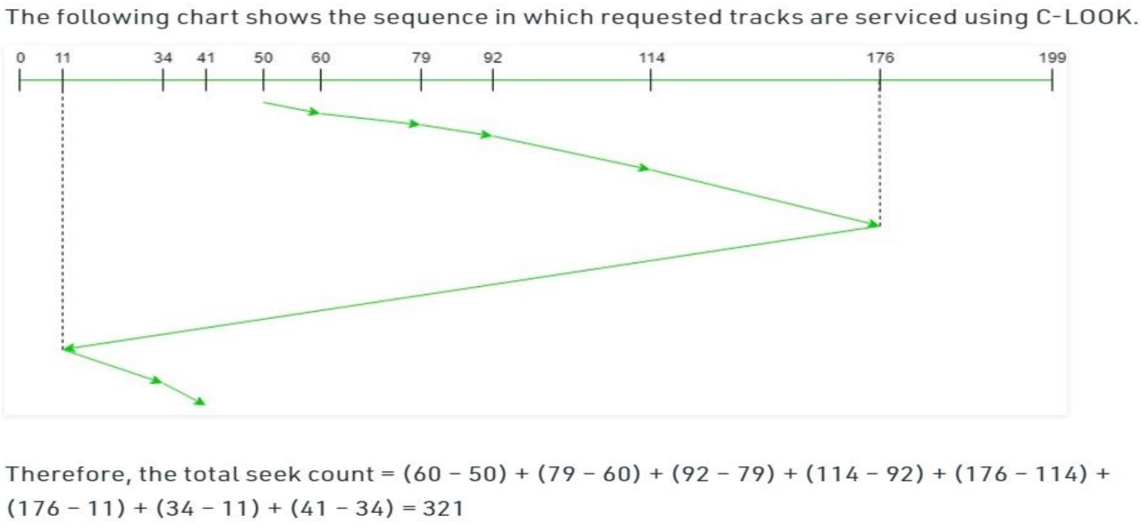
C-LOOK is an enhanced version of both **SCAN** as well as **LOOK** disk scheduling algorithms. This algorithm also uses the idea of wrapping the tracks as a circular cylinder as C-SCAN algorithm but the seek time is better than C-SCAN algorithm. We know that C-SCAN is used to avoid starvation and services all the requests more uniformly, the same goes for C-LOOK.

In this algorithm, the head services requests only in one direction(either left or right) until all the requests in this direction are not serviced and then jumps back to the farthest request on the other direction and service the remaining requests which gives a better uniform servicing as well as avoids wasting seek time for going till the end of the disk.



Examples:

Input:
Request sequence = {176, 79, 34, 60, 92, 11, 41, 114}
Initial head position = 50
Direction = right (Moving from left to right)
Output:
Initial position of head: 50
Total number of seek operations = 156
Seek Sequence is
60
79
92
114
176
11
34
41



The following chart shows the sequence in which requested tracks are serviced using C-LOOK.

Therefore, the total seek count = (60 – 50) + (79 – 60) + (92 – 79) + (114 – 92) + (176 – 114) + (176 – 11) + (34 – 11) + (41 – 34) = 321

Conclusion: Thus we have studied and implemented disk scheduling algorithms: SSTF, SCAN, C-Look considering the initial head position moving away from the spindle.