

Government Engineering College Thrissur

System Software Lab

Navaneeth D

TCR18CS043

S5, CSE

Disc Scheduling Algorithm

AIM

Simulate the following disk scheduling algorithms. *

a) FCFS b)SCAN c) C-SCAN

THEORY

Disk scheduling is done by operating systems to schedule I/O requests arriving for the disk. Disk scheduling is also known as I/O scheduling.

Disk scheduling is important because:

- Multiple I/O requests may arrive by different processes and only one I/O request can be served at a time by the disk controller. Thus other I/O requests need to wait in the waiting queue and need to be scheduled.
- Two or more request may be far from each other so can result in greater disk arm movement.
- Hard drives are one of the slowest parts of the computer system and thus need to be accessed in an efficient manner.

ALGORITHMS :

1. **FCFS**: FCFS is the simplest of all the Disk Scheduling Algorithms. In FCFS, the requests are addressed in the order they arrive in the disk queue. Let us understand this with the help of an example.
2. **SCAN**: In SCAN algorithm the disk arm moves into a particular direction and services the requests coming in its path and after reaching the end of disk, it reverses its direction and again services the request arriving in its path. So, this algorithm works as an elevator and hence also known as **elevator algorithm**. As a result, the requests

at the midrange are serviced more and those arriving behind the disk arm will have to wait.

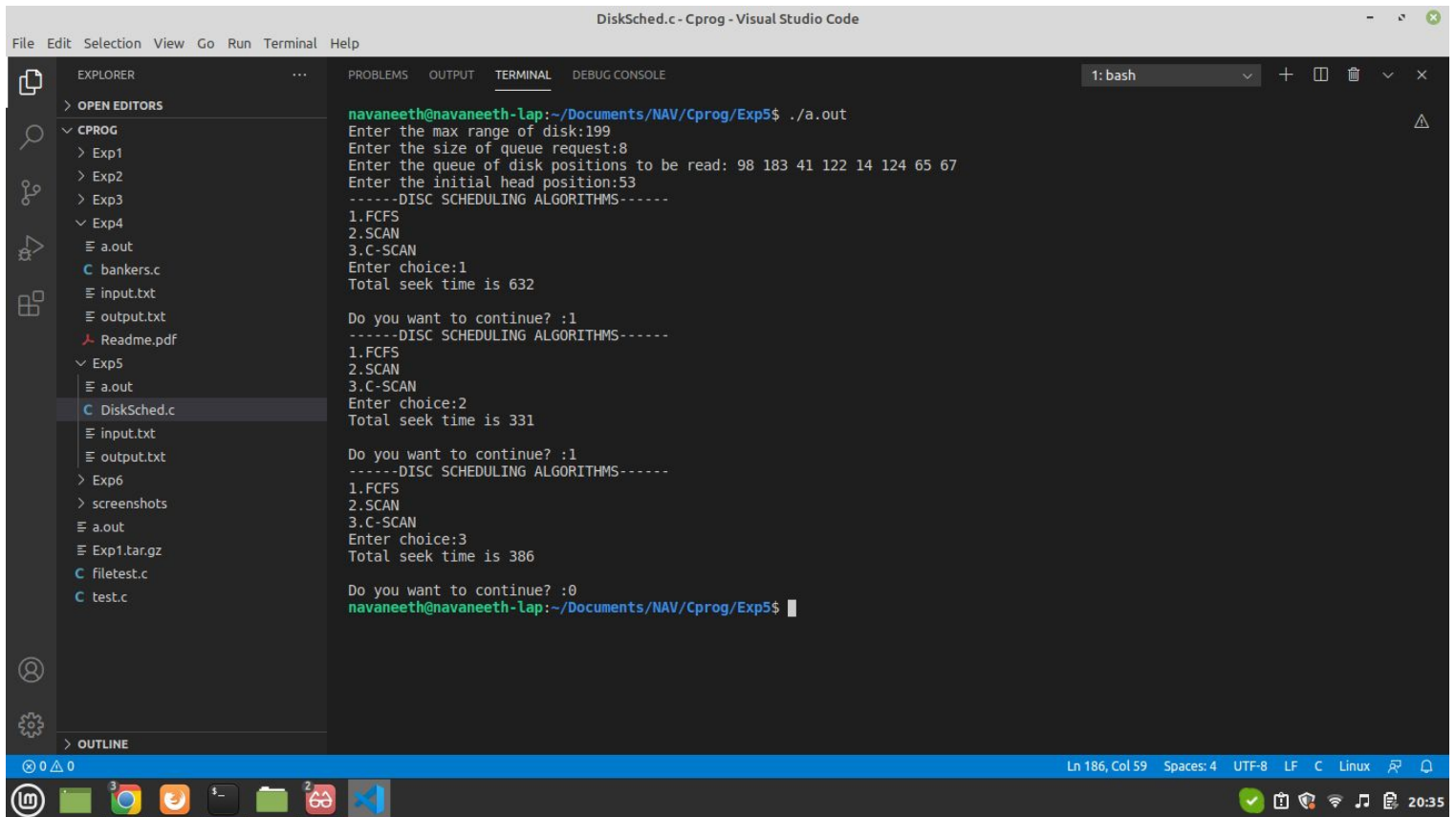
3. **CSCAN**: In SCAN algorithm, the disk arm again scans the path that has been scanned, after reversing its direction. So, it may be possible that too many requests are waiting at the other end or there may be zero or few requests pending at the scanned area.

RESULT

All the Disc scheduling stated above were implemented with successful output.

(next page contains output screenshots)

Output Screenshots



The screenshot shows a Visual Studio Code window with the file `DiskSched.c` open. The terminal window is active, showing the execution of the program `./a.out`. The program prompts the user for several inputs: the maximum range of the disk (199), the size of the queue request (8), and the queue of disk positions to be read (98 183 41 122 14 124 65 67). It then displays the initial head position (53) and the disk scheduling algorithms available (1.FCFS, 2.SCAN, 3.C-SCAN). The user chooses 1 (FCFS), and the program calculates the total seek time as 632. It then asks if the user wants to continue (1), and displays the same menu. The user chooses 2 (SCAN), and the program calculates the total seek time as 331. It then asks if the user wants to continue (1), and displays the same menu. The user chooses 3 (C-SCAN), and the program calculates the total seek time as 386. It then asks if the user wants to continue (0), and the program ends.

```
navaneeth@navaneeth-lap:~/Documents/NAV/Cprog/Exp5$ ./a.out
Enter the max range of disk:199
Enter the size of queue request:8
Enter the queue of disk positions to be read: 98 183 41 122 14 124 65 67
Enter the initial head position:53
-----DISC SCHEDULING ALGORITHMS-----
1.FCFS
2.SCAN
3.C-SCAN
Enter choice:1
Total seek time is 632

Do you want to continue? :1
-----DISC SCHEDULING ALGORITHMS-----
1.FCFS
2.SCAN
3.C-SCAN
Enter choice:2
Total seek time is 331

Do you want to continue? :1
-----DISC SCHEDULING ALGORITHMS-----
1.FCFS
2.SCAN
3.C-SCAN
Enter choice:3
Total seek time is 386

Do you want to continue? :0
navaneeth@navaneeth-lap:~/Documents/NAV/Cprog/Exp5$
```