

CS342 Operating Systems – Spring 2018

Project 4: Disk Scheduling Simulation

Assigned: April 30, 2018

Due date: May 15, 2018, **23:55** (Moodle) (no deadline extension will be granted)

You can do this project as a group of two students each. You have to program in C on Linux.

Objectives: Practice developing a simulator and doing simulation experiments; practice disk scheduling.

Background [please read Section 12.4 of the textbook]: Physically, a disk is a set of concentric cylinders. In each cylinder, there are a lot of blocks. Hence a given block is on some cylinder and access latency to the block is proportional with the distance of the disk head from the current position (current cylinder on which the disk head is) to the cylinder where the block resides. Hence, for magnetic (spinning) disks, access time to disk to retrieve a block (on a cylinder) is affected by the movement of the disk head to the cylinder containing the requested block. If there are several such block requests pending in the disk scheduling queue, these requests can be served by considering the cylinders where they reside. There are disk scheduling algorithms like FCFS, SSTF, LOOK and CLOOK. FCFS serves the requests in first-come first-served manner (according to their order in scheduling queue). SSTF serves the shortest distance request first (the closest request to the current head position). LOOK algorithm scans the disk from one end to the other while serving requests; we do not go till the end of the disk, but to the last request in that direction. Then, direction is reversed and requests are continued to be served in the new direction. CLOOK is a circular version of LOOK and serves the requests only in one direction; for example from left to right. When we come to end (to the right-most), we reverse the direction and go the beginning (to the left-most request) without serving (but still moving the disk and spending time). Then we change the direction again towards right and serve the requests again while going from left to right. At an arbitrary time t , the request queue may contain a sequence of pending requests like: 98, 183, 37, 122, 14. This means, for example, there are 5 requests pending to retrieve 5 blocks. One block is on cylinder 98, another block cylinder 183, etc. Cylinder numbers matter, not the block numbers (we have to move the disk head physically to that cylinder from the current head position).

First read the related section (**Section 12.4 – Disk Scheduling**) from our **textbook**. Then do this project.

In this project, you will develop a program that will simulate the following disk scheduling algorithms: FCFS, SSTF, LOOK and CLOOK (circular look). It will get the input (a sequence of disk requests with their arrival times) from an input file and will print out the following information for each scheduling algorithm: the total time (total head movement in terms of cylinders skipped over) required to serve all requests, the mean request wait time, and the standard deviation of the request wait time. Wait time for a request is the difference between the time the request is started being served and the time the request has arrived to the queue. Assume that rotational latency is 0 and transfer speed is infinite (hence when we are on the correct cylinder, we can access the requested block or blocks on that cylinder immediately). Assume the time required to go from one cylinder to the next one is 1 time unit. Assume there are N cylinders numbered from 1 (innermost – leftmost) to N (outermost - rightmost). N will be given as a parameter to the program.

An example input file is shown below:

```
10 478
18 256
18 389
18 389
27 132
45 90
50 410
```

This means, at time 18, for example, a request for a block on cylinder 256 has arrived (there are other requests arrived at the same time as indicated by the other lines). Note that at each line of input, we have a time of arrival (an integer value) expressed in time units and a cylinder number. The input file is sorted with respect to the arrival time.

An example output (not related to the values above) is shown below (just to give an example for the format):

```
FCFS : 2360 30 60
SSTF : 2146 20 52
LOOK : 1690 32 50
CLOOK: 1912 35 40
```

The output says, for example, that for FCFS scheduling, the total time to serve all the requests is 2360 time units, the average wait time is 30 and the standard deviation of the wait time is 60. Note that the values in the example above are just given randomly (may not be used to compare the algorithms, i.e., may not make sense when compared).

The program executable will be called **ds** and an example invocation of it can be as below:

```
ds <N> <infile.txt>
```

Here, <N> is the number of cylinders on the disk (1..N). <infile.txt> is the input file.

Report. Design and do some experiments with your simulator and write a report about the results. Your report will include plots.

Submission. You will submit your program (a Makefile, a README.txt file, and ds.c) through Moodle. Put everything into a folder (as usual) and do your submission. Include a Makefile. Include a README.txt file (containing the names of students in the group). One submission per group is enough. Late submissions are not accepted. No deadline extension will be provided.

Clarifications:

- Maximum number of requests that may arrive at the same time is M=100.