CS342 Operating Systems – Fall 2018
Project 4: Disk Scheduling

Assigned: Dec 14, 2018
Due date: Dec 29, 2018, **23:55** (Moodle)

You will do this project individually. You have to program in C and Linux. You are recommended to use the following distribution of Linux: **Ubuntu 16.04 – 64 bit**.

Part A: Processes [35 points]
Objective: Practice mass storage, C programming, statistics knowledge.

In this project, you will write a program (diskschedule.c) that implements the following disk scheduling algorithms. a) FCFS; b) SSTF; c) SCAN; d) C-SCAN; e) LOOK; f) C-LOOK.

Your program will service a disk with 5000 cylinders numbered 0 to 4999. It will service 1000 requests according to each of the algorithms listed above. The program will be passed the initial position of the disk head (as a parameter on the command line) and report the total amount of head movement required by each algorithm. The workload (requests) will be given in two ways: 1) will be randomly generated in your program; 2) will be read from an input file. Which one to use will be specified at the command line. The program will be invoked as follows:

diskschedule <headpos> <inputfile>

The name of the program will be **diskschedule**. <headpos> is the initial head position. Assume the initial direction (when required) is always towards right (to bigger cylinder numbers). If <inputfile> is not given, then the requests will be generated randomly. The format of the input file (ascii text file) is given in the example below. Each line is for a different request. A line contains a request number and a cylinder number.

```
1 4000
2 2000
3 4500
4 1450
5 6534
…
999 452
```

An example invocation of the program can be:

```
diskschedule 1230 in.txt
```

The output file be in the following example format:

```
FCFS: 15000
SSTF: 15000
SCAN: 15000
```

```
C-SCAN: 15000
LOOK: 15000
C-LOOK: 15000
```

Part C: Experiments [20 points]
Objective: Practice designing and conducting experiments and applying knowledge and skills acquired in the Probability and Statistics course.

Run the program 100 different times with random input (random requests and random initial head position). At the end, for each algorithm, find out the *average* total movement and *standard deviation* of the total movement. Report the results in a table.

Submission

Put all your files into a project directory named with your ID (one of the IDs of team members), tar the directory (using **tar xvf**), zip it (using **gzip**) and upload it to Moodle. For example, a student with ID 20140013 will create a directory named 20140013, will put the files there, tar and gzip the directory and upload the file. The uploaded file will be 20140013.tar.gz. Include a **README.txt** file as part of your upload. It will have the name and ID of the student, at least. Include also a **Makefile** to compile your program. We want to type just **make** and obtain the executables. Do not forget to put your report (PDF form) into your project directory.

Additional Information and Clarifications

- Additional clarifications can be posted in piazza or on the course website besides project specification.