Programming Assignment 3: Disk Scheduling Algorithms
CSC 4103: Operating Systems, Spring 2020

Due: April 30th, Thursday (by 11:59 PM)
Total Points: 10

Instructions: Compile and test-run your code on the classes server. Submit your work as instructed and verify your submission. The verify command will display your submission date/time. Include your name, email, and classes login ID in all source code files. **Late submissions** will be penalized at the rate of 10% per day late and no more than 3 calendar days late.

How to Submit (Online Class): You need to complete your submission in two steps.
(1) Submit your code to the classes server. In order to connect to the classes server, you will need to use VPN. The following link contains a brief instruction for using VPN: https://www.csc.lsu.edu/~fchen/class/csc4103-sp20/files/VPN.pdf
(2) Submit a simple note on Moodle. This note should include necessary information, such as your name, classes login, and myLSU email.

Note: The UNIX time of your submission on the classes server will be used to determine the exact submission time. So the submission time is the time when you upload your code.

Objective
To implement disk-scheduling algorithms and calculate the disk access times with a simple disk model.

Background
As disk drives are relatively slow devices, an effective disk-scheduling algorithm is important to operating system performance. Various disk-scheduling algorithms exist, such as FCFS, SSTF, SCAN, and C-SCAN, etc. We have learned these algorithms in the class and applied them to a list of disk requests.

Programming Language
C/C++ or Java.

Programming Task
Write a program that implements the FCFS, SSTF, and SCAN disk-scheduling algorithms. The program simulates a simple disk drive, which has a specified number of logical blocks numbered from 0. For example, assuming a disk capacity is 200 blocks, the logical blocks are numbered from logical block address (LBA) #0 to #199. The program receives a sequence of disk requests, runs the specified scheduling algorithm to determine the order of servicing the requests, and calculates the time of servicing each request, and reports the total time.

Detailed Requirements
(1) Your program should accept four parameters on the command line as input:
   a. The current disk head position.

b. The maximum number of logical blocks on the disk.

c. The disk scheduling algorithm (FCFS, SSTF, or SCAN).

d. The input file that contains a sequence of disk requests in their arrival order.

Example: `$ ./disk 53 200 FCFS requests.txt`

The above example command means that the simulated disk has 200 blocks in total (from LBA #0 to LBA #199), the current disk head position is on block LBA #53, uses FCFS scheduling, and takes requests.txt as the input.

Disk Request File Format: Each line contains the logical block address for access. An example input file is available online for your reference.

About SCAN algorithm: We assume the disk head initially moves towards the lower address (the end of LBA #0).

(2) Your program should service the sequence of requests in the order that is decided by the user-specified algorithm.

(3) Each request takes a certain amount of time to complete. The time is the sum of three major components, (1) seek time, (2) rotational latency, and (3) data transfer time. For a given request to block *T*, we use a simple model to calculate the time for moving disk head from current position *C* to the target position *T*.

$$Access\ Time = seek\_time + rotational\_latency + transfer\_time$$

Seek time: $|T - C|*0.1$, where *C* and *T* are the current and the target blocks.
Rotational latency: *rand(0, 1)* * 8.3, where *rand()* returns a random floating-point number in range [0, 1].
Transfer time: 0.031, the time of transferring a block.

(4) For each disk request to target block *T*, the program should generate the following output information upon completion of a request:

a. The target disk head position (*T*).
b. The current disk head position (*C*).
c. The distance between the current and the target head position (|*T*-*C*|).
d. The seek time (|*T* - *C*|*0.1)
e. The rotational latency (*rand(0, 1)*8.3)
f. The transfer time (0.031)
g. The access time (the sum of seek time, rotational latency, and transfer time).

(5) After the entire program completes, output the following information:

a. The total number of disk requests.
b. The total amount of disk head movement (the sum of all distances 4c above).
c. The total amount of seek time (the sum of 4d above)
d. The total amount of rotational latency (the sum of 4e above)
e. The total amount of transfer time (the sum of 4f above)
f. The total amount of access time (the sum of all access times 4g above).

(6) Your submission should include an README file to clearly explain how to compile and run your code, such as the command, parameters, and the expected input and output, and any other necessary information. In the README file, please also provide your full name, LSU ID, and email address.

Submitting Your Work

All files you submit must have a header with the following:

Name:	Your Name (Last, First)
Project:	PA-3 (Disk Scheduling Algorithm)
File:	filename
Instructor:	Feng Chen
Class:	cs4103-sp20
LogonID:	cs4103xx

You need to use the server "**classes.csc.lsu.edu**" to work on the assignment. You can login to your account in the server using SSH. Create a directory **prog3** (by typing **mkdir prog3)** in your home directory, and then you create your program or source code in **prog3**.

Note that do NOT include any directory in **prog3**.

Make sure that you are in the **prog3** directory while submitting your program. Submit your assignment to the grader by typing the following command:

~cs4103_chf/bin/p_copy 3

This command copies everything in your prog3 directory to the grader's account. Check whether all required files have been submitted successfully:

~cs4103_chf/bin/verify 3