Extra credit Lab Assignment (10 points)

Simulating Disk Scheduling Algorithms

Due date: 3:30 pm on December 1 (Friday), 2017

1. Goal of this programming assignment

The primary goal of this lab assignment is to exercise four disk scheduling algorithms and evaluate the effectiveness of each algorithm which affects the performance of disk.

2. Lab assignment description

Write a C or C++ program that simulates the operation of 4 disk scheduling algorithms used in a disk management system:

1. FCFS (First-come-First-serve) algorithm
2. SSTF (Shortest Seek Time First) algorithm
3. SCAN algorithm
4. C-LOOK algorithm

You will simulate each algorithm on the input data sets and report the performance of each. A description of each of these algorithms appears in Chapter 10 of our text.

2.1 Specification

In this lab assignment, you will be calculating the seek time of the each different disk scheduling algorithms for the given disk cylinder reference string for one process only. Your program should accept four arguments, which are the type of algorithm, current cylinder position, the direction of cylinder movement (up (1) or down(0)), the names of an input file, and an output file. The input file contains the disk cylinder reference string in the order of disk access request, and the output file stores the display the cylinder movement and the total cylinder traverse results.

Sample usage is

```
prompt> disksim position direction input output
   where
    1.  position: current disk head position in cylinder (0~199)
    2.  direction: direction of disk head movement – 0 (down) or 1 (up)
    3.  input: input file has serious disk cylinder access request
    4.  output: tracking result of disk arm movements and total number
        of disk arm movements (traversal time)
```

2.2 Input details:

The file contains the disk access request sequence string for only one process. The each line represents a cylinder number which is being accessed by the process. Assume a disk has total 200 cylinders from 0-199. This implies that no number in the reference stream will fall outside that range. There will be no more than 50 references in the disk reference string.

2.2. Output details:

The output file should print out the movement of disk for each request for each disk scheduling algorithm and summary of total cylinder movements after running all four disk algorithms. Sample output format is shown below. For example, the current disk head position is 53, and disk is moving up direction. Note that these are just sample values.

Assume Contents of myinput file is as below:

```
98, 183, 37, 122, 14, 124, 65, 67

Linux prompt> disksim 53 1 myinput myout
```

Contents of myout:

```
================================================================================
[FCFS] Disk Scheduling Algorithm Simulation
Currently disk head at 53 and is moving up
================================================================================
[53->98], [98->183], [183->37], [37->122], [122->14], [14->124], [124->65],
[65->67]

Total number of cylinder movements: 640 cylinders

================================================================================
[SSTF] Disk Scheduling Algorithm Simulation
Currently disk head at 53 and is moving up
================================================================================
[53->65], [65->67], [67->37], [37->14], [14->98], [98->122], [122->124], [124-
>183

Total number of cylinder movements: 236 cylinders

================================================================================
[SCAN] Disk Scheduling Algorithm Simulation
Currently disk head at 53 and is moving up
================================================================================
[53->65], [65->67], [67->98], [98->122], [122->124], [124->183], [183->199],
[199->37], [37->14]

Total number of cylinder movements: 331 cylinders
```

```
================================================================================
[C-LOOK] Disk Scheduling Algorithm Simulation
Currently disk head at 53 and is moving up
================================================================================
[53->65], [65->67], [67->98], [98->122], [122->124], [124->183], [183->14],
[14->37]

Total number of cylinder movements: 322 cylinders
```

3. Programming Requirements

(1) Programming language: You have to use either C or C++ to develop your simulator on Linux OS environment
(2) Running Environment: Your program should be compiled at **csegrid** and be able to be tested without errors.

4. Deliverables

(1) Program source codes includes all source program files, Makefile, and Readme
(2) Sample output

5. Evaluation criteria

(1)	Completeness for each algorithms	10 points
i.	FIFO	1 points
ii.	SSTF	3 points
iii.	SCAN	3 points
iv.	C-LOOK	3 points

6. How to turn in my work

Please do the followings when you submit your programming assignment.

- Create a tar file that contains your written source code, makefile and readme. DO NOT INCLUDE EXECUTABLES AND OBJECT FILES.
- Please use the following convention when you create a tar file
  - First 3 letters of your last name + last 4 digits of your student ID
  - e.g.: If a student name is “Bill Clinton” and his ID is 999-34-5678, then his tar file name is “cli5678.tar”.
- Once you create the tar file, and compress the tar file using ‘gzip’.
  - Do not know how to tar or zip your files?
    - Check “man tar” at Unix prompt
- Upload **your gzipped file to class Canvas.**