Metropolitan Stat University, Saint Paul, Minnesota
ICS 462 Operating Systems
Assignment 6 Part 3

Points: 10

Goal: To learn to program using concurrent data structures and compare disk scheduling algorithms

Problem: In this part of the assignment, you will implement the Look disk scheduling algorithm.

Download the file Disk Scheduling Algorithms.zip from the D2L site. This is a Java Eclipse project. Open the project and execute it. It will ask for the number of requests. Enter any positive number. The code executes, but it is an infinite loop. Kill the process.

Next, examine the code and understand it. In particular, look at the following two files.

Scheduler.java
Look.java

Scheduler.java contains the basic code that is common to all disk scheduling algorithms. It lacks the following:

1) Fields to keep track of statistics
2) The code that goes into the methods initialize(), processNextRequest(), and printStatistics(). The first two are declared abstract.

The class Look extends Scheduler and is supposed to implement the Look scheduling algorithm. The only code you need to write here are:

1) The code for initialize(), which happens before any disk request is serviced.
2) The code for processNextRequest() to process a single request. Rough pseudo code for implementing this method is given in the Java file.

You need to change nothing else in this project. (While you are permitted to change anything else, you will probably end up taking more time to complete the project. Moreover, the user interface and program functionality should be as specified.) The total number of lines to be added should be under 60.

It is *your* responsibility to understand the code, complete it, and do all testing and *debugging*.

When the code is finished, you should get displays of the following kind.

```
Enter number of requests: 100
Look:
Number of tracks moved 1020
Requests Processed 100 time spent 93
```

You need to submit the implementation (the source code for the classes Scheduler and Look, and nothing else) by 11:59 PM on July 23. I will accept late submissions until 11:59 PM on July 24. There will be a 10% penalty for late submissions. I will post my implementation on July 25.