Operating Systems
CS4348

Project #3: OS Scheduling Algorithms

Due Date: Monday, April 23, 2018

I. Project Organization

This project demonstrate six different scheduling algorithms.

You should do the following pieces to complete your project. Each piece is explained below:

Code	60 points
Output	30 points
Summary	10 points

Code

Your code should be nicely formatted with plenty of comments. The code should be easy to read, properly indented, employ good naming standards, good structure, etc.

Output

Output will be graded by running your program. Each algorithm's output is 5 points.

Summary

The summary section should discuss your project experience. You should discuss how you approached the project, including anything difficult or interesting, what was learned, and the end result. This should be at least 1 page in length.

II. Project Description

Language/Platform/Approach

This project must target a Unix platform and execute properly on our CS1 server.
The project must be written in C, C++, or Java.

Problem Overview

This project will simulate a scheduler scheduling a set of jobs.

The project will allow the user to choose a scheduling algorithm from among the six presented in the textbook. It will output a representation of how the jobs are executed.

Design

You may design your own implementation approach, but here are a few constraints.

Your program should read in a list of jobs from a tab-delimited text file named jobs.txt. The format of the text file should have one line for each job, where each line has a job name, a start time and a duration. The job name must be a letter from A-Z. The first job should be named A, and the remaining jobs should be named sequentially following the alphabet, so if there are five jobs, they are named A-E. The arrival times of these jobs should be in order.

The scheduler choice should be a command-line parameter that is one of the following: FCFS, RR, SPN, SRT, HRRN, FB, ALL. If ALL is input, the program should produce output for all six scheduling algorithms. RR and FB should use a quantum of one. FB should use three queues.

Your output should be a graph as shown in the slides. The graph can be text or you can use a graphics package such as JavaFX to draw the graph. For text, you may draw the graph down the page rather than across.

Your program should be able to reproduce the sample shown in the book as well as any similar set of jobs.

Sample Output

Below is sample text-based output. For graphical output, you can make the graph look like the ones in the textbook and slides.

```
FCFS

A  XXX
B     XXXXXX
C           XXXX
D               XXXXX
E                    XX
```

```
FCFS (this is another way you may print the output instead of the one above)

A B C D E
X
X
X
  X
  X
  X
  X
  X
  X
    X
    X
    X
    X
      X
      X
      X
      X
      X
        X
        X
```

III. Project Guidelines

Submitting

Submit your project on eLearning. Include in your submission the following files:

1) readme.txt. A readme file describing how to compile and run your project
2) summary.doc A Word document for the summary
3) Your source files

Academic Honesty

All work must be your own. If cheating is suspected, you will be referred to the Judicial Affairs Office for further discussion. Copying may be detected in a number of ways, including by software which compares your code with all other students' source code, by comparison with code on the Internet, or by a visual inspection of your source code.

Resources

The slides and textbook describe the six algorithms to be used.