CMPE142 : Operating Systems
Assignment 2: Scheduling
Version 2

Due: Apr 30 2019 Before Midnight

In this assignment you will create a program that simulates various schedulers we learned about in class. Your program will read job information from a file, and simulate the scheduling algorithms from the book, outputting statistics for each scheduler.

This assignment is worth up to 20 points and may be done in groups of up to **three people max.** Each team member can receive up to 20 points. It is expected that groups of more than one student will find an equitable way to distribute the work outlined in this assignment. Groups of fewer than three people will still be expected to do the same assignment (eg, no skipping parts if you are working solo).

Prerequisites

- You will need a Linux development environment. We outlined how to create this in class.
- A github repository to store your code (see below)

The Assignment

Your program will read a file called ‘jobs.dat’ containing information about the jobs in the system. The jobs.dat file contains one line for each job in the system. Each line adheres to the following format:

```
JOB_ID      ARRIVAL_TIME      DURATION
```

Between each field above is an arbitrary amount of whitespace (tab, space).

For example, a sample jobs.dat file describing three jobs might look like:

```
0       0       100
1       10      50
2       25      30
```

All fields are numeric, and you may assume no job id will repeat. The maximum number of jobs is 100. Note that this does not necessarily mean only job IDs 0-99 can be used (for example, a job id of ‘562’ is also acceptable).

Start by reading in the jobs.dat file. Once complete, simulate the following scheduling algorithms on the input:

- FIFO (no preemption)
- SJF (no preemption)
- BJF (no preemption)
- STCF
- RR

For each of the above, output the following:

- Start time for each job
- Finish time for each job
- Total time elapsed for each job
- Response time for each job

I don’t have a requirement for the precise output format, but it should be easy to understand.

Start by creating a new github repository for your code. You can sign up for a free github account online if you don’t have one already. Each team member should have their own account, but it does not matter which account owns the repository for this assignment. The repository can be either public or private. Please do not re-use the assignment 1 repository (make a new one for this assignment).

Next, develop the basic framework – reading reading the jobs.dat file, for example. After you have that working, you can split up the remaining work among your team members. It is up to each team to agree on an equitable division of work among the team members.

Teams are **strongly** urged to “commit early, commit often” - this means as soon as you have something working, push your changes to your team’s repository. This ensures that everyone has access to the latest code, and also serves as a backup in case your copy of the code gets messed up.

You may use any programming language you wish.

Submission and Grading

I will use 3 different jobs.dat files as input. For each scheduling algorithm that outputs the correct value (for each jobs.dat file), you will receive 1 point, for a maximum of 15 points. Each team member will be awarded up to 5 points based on the answers to the questions below.

On or before the due date, send answers to the questions below via E-Mail or Canvas, and grant me access to your team’s github repository (my github ID is ‘mlarkin2015’). **Each team member must answer the questions separately.**

The answers to the questions below shall be made via email to the email addresses listed in the class syllabus/green sheet, or via Canvas. DO NOT WAIT UNTIL LATE ON THE DUE DATE, as email server lags or delays may result in a late submission. This is one area that I am extremely picky with – even 1 second late will result in a zero score for that part of the assignment.

I will be comparing all submissions to ensure no collaboration has taken place. Make sure you do not copy another group's work. If you copy another group’s work, members of both groups will receive an F in the class and be reported to the department chair for disciplinary action. If you are working in a group, make sure your partners do not copy another group’s work without your knowledge, as all group members will be penalized if cheating is found.

Make sure your final changes are pushed to github before midnight of the due date. I will be cloning your repository “as of” midnight on that date. This means any changes made afterward will not be considered.

Questions

1. Provide the name of the github repository (including the owning account, eg “mlarkin2015/142-assignment-2”)
2. Provide instructions I need to follow to build your scheduler.
3. What is the name of your scheduler? (Eg, what do I type to invoke it?)
4. For each member in your team, provide 1 paragraph detailing what parts of the lab that member implemented / researched. This may seem obvious but this is the way I use to ensure that each team member was actually contributing usefully to the team. If you worked on the assignment by yourself, you can just send me an email saying “I did the project by myself”, and that will be sufficient.