CSCI 3323 (Principles of Operating Systems), Fall 2020
Homework 3b

Credit: 25 points.

1 Reading

Be sure you have read, or at least skimmed, Chapter 2, section 2.4.

2 Programming Problems

Do the following programming problems. You will end up with at least one code file per problem. Submit your program source (and any other needed files) by sending mail to my TMail address with each file as an attachment. Please use a subject line that mentions the course and the assignment (e.g., "csci 3323 hw 3b" or "O/S hw 3b"). You can develop your programs on any system that provides the needed functionality, but I will test them on one of the department's Linux machines, so you should probably make sure they work in that environment before turning them in.

1. (25 points)

   The starting point for this problem is a Scala script SchedulerStarter.scala that simulates execution of a scheduler, as we did for a simple example in class for the FCFS and SJF algorithms. Comments describe input and desired output. Currently the program simulates only the FCFS algorithm. Your mission is to make it simulate additional algorithms:

   - SJF without preemption.
   - Priority scheduling with no preemption (and assuming that larger numbers mean higher priorities).
   - Round robin, with time quantum as a parameter.
   - Optionally (extra credit), SRTN (which is just SJF with preemption).
   - Optionally (extra credit), priority scheduling with preemption.

   Sample input and output:

   - Simple sample input; output.

   Note that this is a Scala script rather than a program to compile (this seemed simpler to me).

   Comments in the code indicate what you need to fill and some tips for doing so. Note that while this may seem like a daunting assignment just based on the description, I think when you look at the starter code you'll realize that it's not so bad. The two non-preemptive algorithms need not be difficult, and can in fact look a lot alike. Round-robin is harder but doable if you take the approach I recommend.

   NOTE that I discovered in grading that the starter code did not work if there were gaps in the input — i.e., times when all jobs that had arrived had been completed but there was more work arriving later. Embarrassing! I'm not deducting points if your code also doesn't work for such input. If you're interested in a correct version of the starter code, find it in SchedulerStarterCorrected.scala.

3 Pledge

Include the Honor Code pledge or just the word "pledged", plus *at least one of the following* about collaboration and help (as many as apply).[1] Text *in italics* is explanatory or something for you to fill in. For programming assignments, this should go in the body of the e-mail or in a plain-text file `pledge.txt` (no word-processor files please).

- This assignment is entirely my own work. *(Here, "entirely my own work" means that it's your own work except for anything you got from the assignment itself — some programming assignments include "starter code", for example — or from the course Web site. In particular, for programming assignments you can copy freely from anything on the "sample programs page".)*

- I worked with *names of other students* on this assignment.

- I got help with this assignment from *source of help — ACM tutoring, another student in the course, the instructor, etc. (Here, "help" means significant help, beyond a little assistance with tools or compiler errors.)*

- I got help from *outside source — a book other than the textbook (give title and author), a Web site (give its URL), etc.. (Here too, you only need to mention significant help — you don't need to tell me that you looked up an error message on the Web, but if you found an algorithm or a code sketch, tell me about that.)*

- I provided help to *names of students* on this assignment. *(And here too, you only need to tell me about significant help.)*

4 Essay

Include a brief essay (a sentence or two is fine, though you can write as much as you like) telling me what if anything you think you learned from the assignment, and what if anything you found found interesting, difficult, or otherwise noteworthy. For programming assignments, it should go in the body of the e-mail or in a plain-text file `essay.txt` (no word-processor files please).

[1] Credit where credit is due: I based the wording of this list on a posting to a SIGCSE mailing list. SIGCSE is the ACM's Special Interest Group on CS Education.