CSCI 3323 (Principles of Operating Systems), Fall 2022
Homework 2b

Credit: 20 points.

1   Reading

Be sure you have read, or at least skimmed, Chapter 7 of the textbook.

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 putting them in your course "TurnIn" folder on Google Drive. (Note that I want plain-text files, ideally with an extension appropriate for the language — e.g., `.c` for C — but if Google Drive balks at that, rename to have an extension of `.txt`. I want something I can compile as is, except for possibly a change of filename. So no screenshots!) 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. (20 points)

   The starting point for this problem is a Scala script SchedulerStarter.scala that simulates execution of a scheduler, much as the textbook does in Chapter 7, but collecting the information in table form. 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.
   - Round robin, with time slice as a parameter.

   For up to 5 extra points you can make it *also* simulate SJF with preemption.

   Note that the textbook is, I think, a little unclear about what happens in round-robin when a job ends before the end of its time slice. I want you to assume that a new job is scheduled right away, though we will simplify a little bit and assume that everything happens in multiples of some basic time unit, though how that relates to seconds we won't try to say.

   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); if you've forgotten how to run those, it's just `scala` and the source-file name and any parameters (here, the name of the input file and the optional keyword `--verbose`).

   Comments in the code indicate what you need to fill and some tips for doing so. Note that while this may seem like a biggish assignment just based on the description, I think when you look at the starter code you'll realize that it's not so bad. SJF need not be difficult; round-robin is harder but doable if you take the approach I recommend.

3   Essay and pledge

Include with your assignment the following information.

For programming assignments, please put it a separate file. (I strongly prefer plain text, but if you insist you can put it in a PDF — just no word-processor documents or Google Drive links please.) For written assignments, please put it in your main document.

3.1   Pledge

This should 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). Text *in italics* is explanatory or something for you to fill in; you don't need to repeat it!

- I did not get outside help *aside from course materials, including starter code, readings, sample programs, the instructor.*

- 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, 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.)*

3.2   Essay

This should be 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 interesting, difficult, or otherwise noteworthy.