FLORIDA POLYTECHNIC UNIVERSITY

Reading and Simulation Assignment 2

READ CHAPTERS 8 AND 9 TO ANSWER THE FOLLOWING QUESTIONS

Goal: Get a better understanding of process scheduling and scheduling metrics

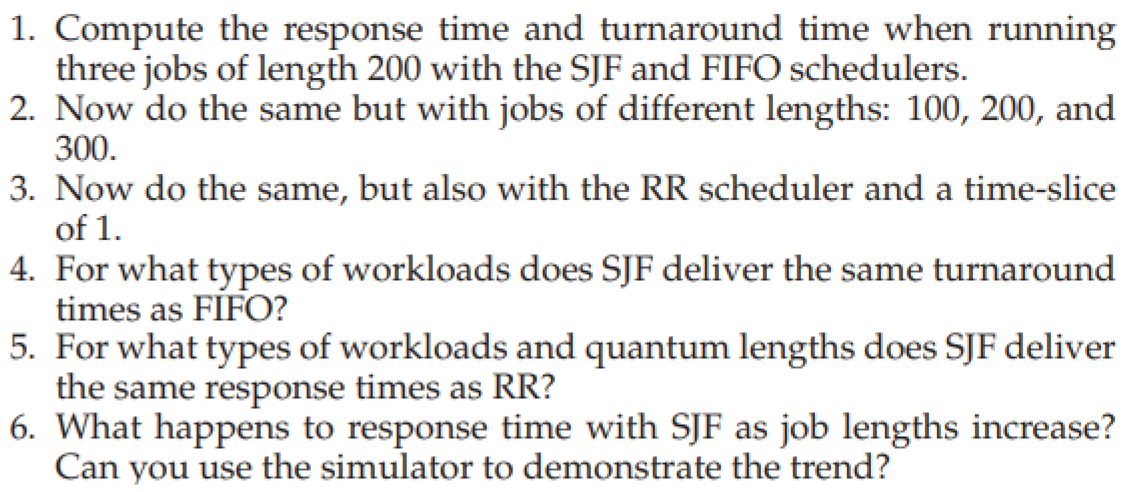
Submission deadline September 24 at 11:59 (Canvas)

Deliveries: A report. The report includes screenshots of the outputs and your description and conclusions about what is happening.

This assignment includes chapters 8, and 9

This program, scheduler.py, allows you to see how different schedulers perform under scheduling metrics such as response time, turnaround time, and total wait time. See the README for details.

From chapter 8



1. Compute the response time and turnaround time when running three jobs of length 200 with the SJF and FIFO schedulers.
2. Now do the same but with jobs of different lengths: 100, 200, and 300.
3. Now do the same, but also with the RR scheduler and a time-slice of 1.
4. For what types of workloads does SJF deliver the same turnaround times as FIFO?
5. For what types of workloads and quantum lengths does SJF deliver the same response times as RR?
6. What happens to response time with SJF as job lengths increase? Can you use the simulator to demonstrate the trend?

From chapter 9

This program, mlfq.py, allows you to see how the MLFQ scheduler presented in this chapter behaves. See the README for details.

1. Run a few randomly-generated problems with just two jobs and two queues; compute the MLFQ execution trace for each. Make your life easier by limiting the length of each job and turning off I/Os.
2. How would you run the scheduler to reproduce each of the examples in the chapter?
3. How would you configure the scheduler parameters to behave just like a round-robin scheduler?
4. Craft a workload with two jobs and scheduler parameters so that one job takes advantage of the older Rules 4a and 4b (turned on with the `-S` flag) to game the scheduler and obtain 99% of the CPU over a particular time interval.
5. Given a system with a quantum length of 10 ms in its highest queue, how often would you have to boost jobs back to the highest priority level (with the `-B` flag) in order to guarantee that a single long-running (and potentially-starving) job gets at least 5% of the CPU?