Process Scheduling

Laboratory 3
Duration: 2 weeks

This assignment will help us learn different process scheduling algorithms and their relative pros and cons.

To do this task, you will need to develop a simulator of a scheduler in C / C++. The simulator must take in the following command line arguments: <scheduling-algorithm> <path-to-workload-description-file>. The simulator must produce as output the following metrics: Turnaround Time (average and maximum) and Run Time of your simulator (not counting I/O). Also, report the schedule in the given format (see the file `schedule_format.txt` : it corresponds to the Preemptive Shortest Job First algorithm applied to `process2.dat`).

For all the studies, we will use the workload description files given here. Each row in the file refers to one process. The row format is as follows:
<process-arrival-time> <cpu-burst-1-duration> <io-burst-1-duration> <cpu-burst-2-duration> <io-burst-2-duration> … -1

For example:
0 100 2 200 3 25 -1 indicates arrival time = 0; CPU burst 1 duration = 100; I/O burst 1 duration = 2; CPU burst 2 duration = 200; I/O burst 2 duration = 3; CPU burst 3 duration = 25; end of process.

Assume that every line ends with -1. A process may have any number of CPU / I/O burst cycles terminated with a -1. There will be any number of processes, terminated by an end of file. The arrival times are in nondecreasing order.

Part I

Implement the following algorithms:

A. First In First Out
B. Non pre-emptive Shortest Job First
C. Pre-emptive Shortest Job First
D. Round robin: experiment with different values of time quantum

Tip: start by designing a nice data structure that captures both the input process description (CPU burst and I/O burst information from the file) as well as the runtime process description (when was it scheduled, how much work is done, etc.). Then design ready and waiting queues, as well as a nice structure to maintain the schedule that you can then output at the end of the simulation.

Part II

Now, suppose you have two processors. Re-evaluate all three algorithms.

Submission:

- Source code with suitable makefiles. The code must output the schedule as well as the statistics.
- Report containing observations in the form of graphs and their analyses.