

## CS 520-A: Introduction to Operating Systems

### The Simulation Programming Project (due November 14, 2016)

(As announced, this project's score will have separate weight of 20% rather than be part of the homework average.)

Please read the remaining slides (47-54) of Lecture 2 discussed in the class, and solve the problems below. (Please remember to document your solutions for the problems you have solved as clearly as possible. For the programming assignments, you must submit the hard copies of both your well-commented code, the output, and the report.)

1. Recall the Book problems 5.3 and 5.12, which you have solved “by hand” in an earlier. Solve them now by writing the event-driven code, as explained in class. Your program must produce as an output the corresponding Gantt charts. (**20 points**)
2. Write a simulation of a (rather simplified) system of Figure 1. You should re-use the core code of the “bus problem” as well as the code you have written for solving Problem 2. Create 10 jobs of random execution time with lengths **uniformly** distributed between 2 and 4 minutes (use the uniform distribution method for random number generation).
  - For each job, the times between I/O requests (i.e, CPU bursts) are distributed **exponentially**. The mean inter-I/O intervals for the jobs are respectively *30ms*, *35ms*, *40ms*, *45ms*, *50ms*, *55ms*, *60ms*, *65ms*, *70ms*, and *75ms*.
  - Each time an I/O is needed it takes precisely *60 ms*
  - A job, once it enters the system, can be either in the *Ready* queue, or *I/O Waiting* queue, or it is being executed by the CPU.

The task is to write the simulation of the system behavior for the whole period of the execution, while computing and collecting the following statistics: *CPU utilization*, *throughput*, *turnaround time*, and *waiting time*.

- a) Different simulation runs are to be performed with the *First-Come-First-Served* and *Shortest-Job-First* (SJF) algorithms. When using SJF, you should experiment with exponential averaging. Use the  $\alpha$  values of 1 (no prediction),  $\frac{1}{2}$ , and  $\frac{1}{3}$ , and observe the effect on the results. (You can experiment with other values, too!)

- b) Think about the conditions under which average waiting time increases with decreasing quantum in round robin and under what conditions it decreases with decreasing quantum. Write a statement describing these conditions. (Note: There is no need to use a quantum that is larger than the maximum CPU burst [computed on the set of all the processes] since in this case RR reduces to FCFS.)
- c) Design experiments to test your statements in Part b). In the first experiment, the processes will satisfy the conditions you listed and will cause the average waiting time to increase with decreasing quantum. In the second experiment, the processes will satisfy the condition you listed that will cause the average waiting time to decrease with decreasing quantum. Each experiment should have about 5 runs which differ only in the value of the quantum. Try to design experiments with the following characteristics:
- At least 10 processes are used, each having many CPU bursts.
  - The largest and smallest CPU burst averages differ by a factor of at least 2.
  - The CPU utilization is between 50 and 90 percent.

Run the experiments using the simulator. For each experiment, create a log file containing the tabular data and Gantt charts for the runs.

You should submit the following:

1. Source code listing;
2. Logs for all runs, as described above; and
3. Report with your observations.

**(80 points)**

**Figure 1**

