CS 4345 (Operating Systems)

Assignment – 2 [Spring 2019]

Due date: 11:00 p.m., Tuesday, March 12, 2019

Purpose: Write a program simulating a simple CPU scheduler to calculate average waiting time.

Problem: Initial ready queue has 5 processes where each process has the following attributes:
i) an identifier (value between 0 and 10, randomly assigned – however, no two processes can have the same id); ii) a burst length between 20 – 100, randomly assigned when it is created; iii) a priority between 1 and 10 (low value indicates higher priority), randomly assigned when it is created (two or more processes may have same priority rank).

Requirement: Display initial snapshot of the system as follows:
Process ID | Priority | Burst-length

Allow user of the program to enter attributes of another process (make sure to have proper check for duplicate process ID). If duplicate ID entered, allow user to enter a correct (unique) process ID.

Display updated snapshot of the system in the same format as above.

Assuming all processes are ready for scheduling, calculate individual waiting time for each process and average waiting time, using the following algorithms: (i) Non-preemptive SJF, (ii) Non-preemptive priority, and (iii) round robin with time quantum 20.

Display the result as follows:
Process ID | Priority | Burst-length | Scheduling algorithm | Total waiting time

Based on your snapshot and calculations, also display average waiting time for each algorithm and order the algorithms from lowest to highest average waiting time.

Notes:

- You may work with absolute values ignoring the time unit. That is, a process may have burst length 46 – you may ignore whether it is 46 seconds or 46 milliseconds.

This program may not need multithreading. Simple data structures (arrays, lists, etc.) should work.

Submission (*this is an individual assignment***):**
You can use Java (preferred) to solve the problem. Include your name, course name, semester, and assignment identifier (Assign 2) as program comments at the top of your code. Submit the source file(s) through BlazeVIEW submission box. **You also need to submit a 'readme' type file including the narrative detailing the thoughts related to the logic and design of your code.**