CS 370 Fall 2016

Programming Assignment 4: Simulating CPU Scheduling Algorithms draft v10/13/2016 3:00 PM

Deadline Oct 27, 5 PM. *Late deadline with penalty Oct 29, 5 PM*

Purpose: The objective of this assignment is to become familiar with some of the CPU scheduling algorithms. You will be implementing three scheduling algorithms some with support for preemption while tracking measures of scheduling effectiveness.

In this assignment you will be implementing three basic CPU scheduling algorithms. Information about the processes that must be scheduled including the number of processes, their start times, and burst durations are provided in a separate file. You will profile the performance of these algorithms by tracking these measures of effectiveness: average turnaround time, average waiting time and throughput.

Description of assignment:

Implementing the following CPU scheduling algorithms using Java with the assumption that all jobs are CPU bound i.e. they do not block for I/O:

- First Come First Serve (FCFS)
- Shortest Job First (SJF) Preemptive
- Round Robin with a given time slice

Profile the performance of these algorithms by tracking these measures of effectiveness average turnaround time, average waiting time, and throughput.

- The turnaround time for a process is the difference between a job's submission and completion times. The average turnaround time reports the average for the set of processes that were scheduled.
- The waiting time for a process reflects the total amount of time spent by that process in the ready queue. The average waiting time reports the average for the set of processes that were scheduled.
- The throughput for a scheduling algorithm measures the total number of tasks processes per unit of time.

Requirements of Task:

1. You need to read the list of processes for your scheduling algorithms from a file. Every line in this file includes a record with comma separated fields. The format for this record is the following: <ProcessID>,<Arrival Time>, <Burst Duration>. All processes in your input files will be provided a unique process ID. There is no limit to the number of the entries. The arrival times and burst durations are integers.

2. Your program (scheduler.java) should support these options: FCFS, SJF preemptive, Priority scheduling, and exit.
3. Implement FCFS and report sequence of execution (begin time and end time for each process CPU burst), average turnaround time, average waiting time, and throughput.
4. Implement SJF (preemptive) and report sequence of execution, average turnaround time, average waiting time, and throughput.
5. Implement Round Robin and report sequence of execution, average turnaround time, average waiting time, and throughput.

Command line arguments for grading:

```
> java scheduler <input_filename>  <quantum>
```

The input_filename contains information about the processes that need to be scheduled.

The quantum (integer) is used to determine the time quantum for round robin.

Submission: Use Canvas to submit a single .tar file named PA4.tar that contains:

- All files related to the assignment (please document your code),
- A README.txt file containing a description of each file and any information you feel the grader may need.

Grading: The grading will also be done on a 100 point scale. The points are as follows: 20 points each for each of the 5 tasks.

Example Output: see here (to be created)

Notes:

1. You are required to work alone on this assignment.
2. Late Policy: There is a late penalty of 10% per-day for up to a maximum of 2 days.