

## COSC 201 – Assignment #2

### Fall 2010

**Objective:** Write a program that will simulate a classic operating system job scheduler.

An OS Job Scheduler takes a job (a process, if you will) that has a certain number of cycles required, and schedules that job to be run. If the jobs come in one at a time and slowly, this is very easy (just schedule them in the order received), however, modern operating systems frequently have to schedule many jobs, with diverse time requirements at the same time.

Your task is to create a class that will simulate a job scheduler. You should research and choose an algorithm for this task – one will not be provided to you in class. You will be graded on the efficiency of your scheduler, both in space and time requirements, on a variety of tests. Keep in mind that there may be very efficient algorithms out there for scheduling but they may be complex. Your primary task is to get a functioning scheduler built. Your secondary task is to make it as efficient as possible.

Each job will have three pieces of data attached: a name, a time requirement (in cycles) and a priority from 0 to 4 (inclusive) with lower numbers receiving higher priority in the scheduler. It is your responsibility to figure out how to deal with priority for this project.

Create a main that will, depending on input, ask for a number from the user then “spawn” that many jobs with random time requirements or take in a list of jobs via file called **Jobs.txt**. The lines in the file will be formatted **jobname, time required, priority**. You will not be required to handle any errors due to the contents of the file. These jobs will be added to a container type of your choosing and you can choose how you store the information for each job. Your main should print out the following information (assume that you will start with cycle 0):

When each job finishes, and when each job switches from running to waiting.

Average completion time over all jobs.

Average wait time over all jobs.

Average completion time over all jobs in priority level 0

Average wait time over all jobs in priority level 0

Average completion time over all jobs in priority level 1

Average wait time over all jobs in priority level 1

Average completion time over all jobs in priority level 2

Average wait time over all jobs in priority level 2

Average completion time over all jobs in priority level 3

Average wait time over all jobs in priority level 3

Average completion time over all jobs in priority level 4

Average wait time over all jobs in priority level 4

You must complete this project in less than 200 lines of code. Comments are not included.

**Expectations:** Your code will need to be neat, concise, well documented and above all, correct (see Testing). All classes should have headers and each method should have comments describing the method's function including javadoc. Any novel or possibly confusing code should be explained, as I do get confused and distracted easily. **Be sure to cite your source for the algorithm you use in this project.**

**Design Documents:** UML diagrams are not required for this project. If you want to create one and have me look it over, I would be happy to do so.

Grading rubric will be given out at least a week ahead of the due date.

**DUE:** November 10, 11:59 pm Eastern.