# CS 211 Programming Assignment #4

Your organization has purchased a new parallel computer (or "cluster") which has several processors. Your task is to design and implement a simple shortest-job-first scheduler that allows multiple users to access the cluster at the same time, as per the following specification:

The scheduler performs all required functions at regular intervals, often called "ticks." During each tick, the scheduler must take into consideration several pieces of information:

1. Are there jobs in our input file that contains a list of jobs to be run? If so, add that job to our list of current jobs. Note that our jobs file contains one job per line in the following format: "<JOB\_NAME> <CPUS\_REQUIRED> <DURATION>". Only one job may be added to the jobs queue during any given tick. If a job has a name of "NULL," skip that job for the current tick (i.e. no jobs will be added during this tick). If a job requires more CPUs than the cluster contains, you should output an error and terminate execution of the program.
2. Are there jobs in the jobs queue? If so, find the job with the shortest duration. Are there enough CPU resources for this job to run? If so, begin execution (i.e. add to the active jobs queue). If, after adding the previous job, are there enough resources to run the next job in the queuejob? If so, go back to the beginning of step #2. Stop adding to the list of active jobs when there are not enough CPU resources available for the job to execute. Try again in the future.
3. For each executing job (note that there may be multiple concurrent jobs), perform one tick's worth of work. After, check to see if the job is done executing. If complete, remove from the active jobs queue and allocate the job's CPU resources back into the resource pool. Note that it is possible for a job with a duration of 1 to be scheduled, executed, and completed during the same tick.

Upon completion, your program should produce a CSV file called "result.csv" that documents the point at which each job enters the jobs queue, finishes its execution, and the difference between these two values.

## Sample Jobs File

Below is a sample jobs file that you can use for testing:

|  |
| --- |
| 12  J1 8 10  J2 2 1  J3 12 12  J4 10 2  J5 5 8  J6 4 2  J7 4 6  J8 2 5  J9 4 3  J10 6 2 |

## Sample Output

Included with this document is the file "output.txt" that contains the output of my program running the sample jobs list with a CPU count of 12 (as specified in the first line of the input file). Also included is the generated CSV file.

## Header Comment, and Formatting

1. Be sure to modify the file header comment at the top of your script to indicate your name, student ID, completion time, and the names of any individuals that you collaborated with on the assignment.
2. Remember to follow the basic coding style guide. A basic list of rules is included with this document.

## Deliverables

You must upload your program and reflection as a ZIP file through Canvas no later than midnight on Monday, October 24, 2016. Remember that your submission must either contain a CodeBlocks or Visual Studio project file!

# Reflection Essay

In addition to the programming tasks listed above, your submission must include an essay that reflects on your experiences with this homework. This essay must be at least 350 words long. Note that the focus of this paper should be on your reflection, ***not*** on structure (e.g. introductory paragraph, conclusion, etc.). The essay is graded on content (i.e. it shows deep though) rather than syntax (e.g. spelling) and structure. Below are some prompts that can be used to get you thinking. Feel free to use these or to make up your own.

* Describe a particular struggle that you overcame when working on this programming assignment.
* Conversely, describe an issue with your assignment that you were unable to resolve.
* Provide advice to a future student on how he or she might succeed on this assignment.
* Describe the most fun aspect of the assignment.
* Describe the most challenging aspect of the assignment.
* Describe the most difficult aspect of the assignment to understand.
* Provide any suggestions for improving the assignment in the future.

## Grading

Your grade will be determined as follows:

* [10] Your program uses a priority queue to model the scheduler.
* Your program uses good Object Oriented Programming principles. As evidence of this, your program uses classes to represent:
  + [5] A given job
  + [5] A CPU within the cluster
  + [5] The cluster of CPUs itself
* [10] Your reflection essay satisfies the requirements as specified earlier in this document.
* [5] Your code is well documented and generally easy to read.
* [60] Does the code compile and run successfully on my test cases? Does the generated CSV file appear to be correct?