CECS 326 **Homework 9** **Fall 2019**

Purpose: This assignment is to gain more experience with scheduling.

You will build a program that simulates round robin scheduling.

Input: You will be given a series of number pairs. Each pair of numbers will represent a process. The first number will represent the arrival time, the second number will represent the processing time. For example, the pairs (0 15) and (5 10) mean that process 0 arrives at time 0 and will take 15 time units to complete; while the process 1 will arrive at time 5 and take 10 time units to complete.

Output: a round robin schedule where the time slice is 5 time units. The output should consist of a time slice number followed by the process running in that time slice or the word "idle" if there is no proces to run.

For example, with the two pairs given above. Your output would be

```
Time   Action
0      Process 0
5      Process 0
10     Process 1
15     Process 0
20     Process 1
25     idle
```

Simplification: all numbers you will be given are multiples of 5.

Numbers: A list of numbers is found in `/net/326/processes.txt`. The numbers are given one pair per line.

Discussion:

Read in the data. (I give you code to do this, see below.)

The current time should start at 0.

Build a time loop. Check if any processes arrive at the current time. If they do arrive, put thier PID (process ID) numbers into the queue. If the queue size is 0, print "idle" otherwise: Get the first PID in the queue; Print the appropriate message; Decrement the timeleft for this process by 5; If the time left is greater than 0, put the PID back into the queue. Increment the current time by 5 at the end of this loop. If there are no processes in the queue AND no processes left to be added to the queue; you need to quit the loop.

Code:

A program that will read the numbers into an array is supplied. It is found in `/net/326/scheduler.c` It also prints the data so you can see what got stored. You can leave the print in when you develop your solution. I use an array of structures to make it easier for you to use my code to implement your program. The PID of a process is equal to it's location in the array. (So you can find it easily.)

A queue implementation is supplied. Get `/net/326/IQueue.h` and `/net/326/IQueue.c`.

Demo: Your scheduler program. Leave a copy of it on the server in your home directory, (do not change the code file name this time). The instructor may test your program with alternate inputs.

Nathan Pickrell 1 October 2019 (Week 6 Lecture 1) Due: 3 October 2019 (Week 6, Lab 2)