COP4600 – Operating Systems – Spring 2015 – Programming Assignment

Learning Objective: To gain a fuller understanding of process scheduling algorithms by implementing them.

Development: Use C, C++, C# or Java. Create a console-mode application that does not use a windowing environment. Document your code adequately, including your name and the course name at the top of every file.

Assignment: Implement the First-Come First-Served, preemptive Shortest Job First, and Round-Robin algorithms as for single processors.

Input: Your program will read a file from the current directory called **processes.in**, which will be formatted as follows. Your program should ignore everything on a line after a # mark and ignore additional spaces in input.

```
processcount 2      # Read 5 processes
runfor 15           # Run for 15 time units
use rr              # Can be fcfs, sjf, or rr
quantum 2           # Time quantum – only if using rr
process name P1 arrival 3 burst 5
process name P2 arrival 0 burst 9
end
```

Note that the processes do not need to be specified in order of arrival, and do not need to have similar names.

Output: Generate a file called **processes.out**, formatted as follows.

```
2 processes
Using Round-Robin
Quantum 2

Time 0: P2 arrived
Time 0: P2 selected (burst 9)
Time 2: P2 selected (burst 7)
Time 3: P1 arrived
Time 4: P1 selected (burst 5)
Time 6: P2 selected (burst 5)
Time 8: P1 selected (burst 3)
Time 10: P2 selected (burst 3)
Time 12: P1 selected (burst 1)
Time 13: P1 finished
Time 13: P2 selected (burst 1)
Time 14: P2 finished
Time 14: Idle
Finished at time 15

P1 wait 5 turnaround 10
P2 wait 5 turnaround 14
```

Clarifications

This version of Round-Robin **should not** run the scheduler immediately upon the arrival of a new process, unless the CPU is currently idle.

Your program **will not** be given an input that results in an ambiguous decision, such as identical arrival times for Round-Robin or identical burst lengths for SJF; you should avoid generating an error in that case on general principles but it will not appear in either the example inputs or the grading inputs.

Submitting

Zip up your code, and upload it to Webcourses. If you have a single source file then you can simply submit it. If you have multiple source files, include an appropriate makefile, ant build file, or project file.