CS431 — Project 2

January 30, 2017

Due: Friday, February 10, 2017 (50 points)

Description

In this project you will be simulating several process scheduling algorithms on lists of processes provided in input files. I will assume you are implementing this in Java but you can use a different language. If you use a different language, I need complete compilation and testing procedures or a working Makefile. If using Java, your program should compile and run with the following:

```
$ javac ProcessScheduler.java
$ java ProcessScheduler input.csv
```

The input file will have one process per line such as:

```
1,100
2,100
3,150
4,100
5,200
```

The first number is the pid and the second number is the total number of cycles the process will execute.

In your program, you should read all processes from the input file then simulate scheduling them until they are all complete. Your program should implement the following schedulers:

1. First-come, first-served scheduler: this scheduler simply runs each process to completion in the order they are submitted.
2. Shortest first scheduler: this scheduler will run each process to completion in the order of shortest to longest. For processes of the same length, execute in order of submission.
3. Round robin scheduler: this scheduler will go through the processes in order of submission executing each for a small amount of time (quantum) before switching to the next process. If a process has less than the quantum in cycles remaining to execute, make sure your total cycles elapsed only goes up by the amount remaining instead of the full quantum.

   Run your round robin scheduler twice: once with quantum 50 and once with quantum 100.
4. Random scheduler: this scheduler will randomly choose a process to run for a quantum weighted by how many cycles are remaining for each process.

   Run this scheduler once with a quantum of 50.

   As an example, consider the following input file:

   ```
   1,100
   2,200
   ```

The first process has 100 cycles and the second process has 200 cycles to execute. Therefore, there is a $\frac{100}{300}$ chance the first process is chosen and a $\frac{200}{300}$ chance the second process is chosen to run.

Suppose our quantum is 50 cycles and the scheduler chooses process 1 to execute. After this, process 1 has 50 cycles remaining to execute and process 2 still has 200 cycles. Now, there is a $\frac{50}{250}$ chance for process 1 to be chosen and a $\frac{200}{250}$ chance for process 2 to be chosen.

When a process finishes executing, your program should print out a line such as:

```
Process 3 finishes on cycle 580.
```

where the cycle number is the overall cycle number from the start of the scheduling. Additionally, when all processes have finished you should print a line stating:

```
Average turnaround time: 250.50.
```

which will show the overall average turnaround time in cycles for all processes.

Repeat this output for all schedulers listed.

Sample Output

Here is sample output for the input file shown above. Note that the random scheduler's output will change each time the program is executed.

```
Running First-come, first-served scheduler.
Process 1 finishes on cycle 100.
Process 2 finishes on cycle 200.
Process 3 finishes on cycle 350.
Process 4 finishes on cycle 450.
Process 5 finishes on cycle 650.
Average turnaround time: 350.00.

Running shortest first scheduler.
Process 1 finishes on cycle 100.
Process 2 finishes on cycle 200.
Process 4 finishes on cycle 300.
Process 3 finishes on cycle 450.
Process 5 finishes on cycle 650.
Average turnaround time: 340.00.

Running round robin scheduler with quantum 50.
Process 1 finishes on cycle 300.
Process 2 finishes on cycle 350.
Process 4 finishes on cycle 450.
Process 3 finishes on cycle 550.
Process 5 finishes on cycle 650.
Average turnaround time: 460.00.

Running round robin scheduler with quantum 100.
Process 1 finishes on cycle 100.
Process 2 finishes on cycle 200.
Process 4 finishes on cycle 400.
Process 3 finishes on cycle 550.
```

```
Process 5 finishes on cycle 650.
Average turnaround time: 380.00.

Running random scheduler with quantum 50.
Process 1 finishes on cycle 350.
Process 2 finishes on cycle 400.
Process 4 finishes on cycle 450.
Process 5 finishes on cycle 500.
Process 3 finishes on cycle 650.
Average turnaround time: 470.00.
```

Submission

1. On https://codebank.xyz, create a project named `CS431-P2`. Follow this naming convention precisely including case.
2. Create your own local repository by the following:
   (a) Navigate to your local directory for this project.
   (b) Run `git init` to initialize the repository.
3. Make sure you add the reference to the remote repository in your local repository with:

   ```
   $ git remote add origin https://codebank.xyz/username/CS431-P2.git
   ```

4. Your project should have a main class named `ProcessScheduler` in a file named `ProcessScheduler.java`. You can have other files or classes, but it should successfully compile and run by simplying using:

   ```
   $ javac ProcessScheduler.java
   $ java ProcessScheduler input.csv
   ```

I will create additional test input files when grading the assignments.

You will lose points if your submission is not correct (e.g., incorrect repository name, file names, class names, or package declaration that causes the above commands to fail to run).