**CS 440 – Operating Systems**
**Project 3 – Process Scheduling Algorithms**
**100 Points (+5 extra credit points) possible**
**Due in Canvas by 11:59 PM on Tuesday, March 22, 2022**

## Instructions

In this project, you will implement a comparison of different process scheduling algorithms: FCFS, SJF, RR, and Random. The user will be prompted for the following information:

- **Seed –** seed value for the random number generator
- **Number of processes** – this is the number of processes to model. Can be an integer between 2 and 10, inclusive.
- **Last possible arrival time** – this is the time the latest process arrives in the ready queue. It is an integer between 0 and 99, inclusive. If 0 is entered, all processes arrive at the same time in the ready queue.
- **Maximum burst time –** this integer indicates the *maximum* possible burst time for a process. Since the processes will have burst times that are randomly determined, this indicates the maximum value for the random number generator to choose from. Should be an integer between 1 and 100, inclusive.
- **Quantum –** for RR processes, this is the quantum size, *q*. Acceptable values are integers between 1 and 100, inclusive.
- **Latency –** the amount of combined time needed to switch a process on and off the CPU, measured in cycle times. Acceptable values are integers between 0 and 10, inclusive.

We will not model priority-based scheduling in this assignment, so assume all processes have the same priority.

If two or more processes arrive at the same time, one is randomly selected (except for SJF)

If two or more processes have the same burst time required, one is randomly selected for SJF.

Your program will prompt the user for the above 6 items, then calculate and output information and calculate several metrics for each algorithm. Ensure outputs are formatted as shown in the example below.

- Average waiting time – for simplicity, we will just calculate the wait time for each process as (time a process enters its *final* burst cycle) – (time process entered the ready queue)
- Average response time
- Time taken for completion of all tasks

```
Enter a seed value: 229
Enter number of processes (2, 100): 4
Enter last possible arrival time (0, 99): 10
Enter max burst time (1, 100): 12
Enter quantum size (1, 100): 5
Enter latency (0, 10): 2

4 processes created.
P     Arrival     Burst
1     0           4
2     3           11
3     8           9
4     4           4
```

For the random process, it randomly selects the next available process from those at time, t). So from the example above, at t= 0, it would run $P_1$, at t=4, when $P_1$ is done, it would randomly choose from $P_2$ or $P_4$, etc. This information needs to be displayed on the screen. For simplicity, there is no need to provide context switch time that elapses as the first or last process.

Random:

```
@t=0, P1 selected for 4 units
@t=4, context switch 1 occurs
@t=6, P4 selected for 4 units
@t=10, context switch 2 occurs
@t=12, P2 selected for 11 units
@t=23, context switch 3 occurs
@t=25, P3 selected for 9 units
@t=34, all processes complete
Completed in 34 cycles.
Avg wait time = ((0-0)+(6-4)+(12-3)+(25-8))/4 = (0+2+9+17)/4 = 28/4 = 7.00
Avg resp time = ((0-0)+(6-4)+(12-3)+(25-8))/4 = (0+2+9+17)/4 = 28/4 = 7.00
```

FCFS:

```
@t=0, P1 selected for 4 units
@t=4, context switch 1 occurs
@t=6, P2 selected for 11 units
@t=17, context switch 2 occurs
@t=19, P4 selected for 4 units
@t=23, context switch 3 occurs
@t=25, P3 selected for 9 units
@t=34, all processes complete
Completed in 34 cycles.
Avg wait time = ((0-0)+(6-3)+(19-4)+(25-8))/4 = (0+3+15+17)/4 = 35/4 = 8.75
Avg resp time = ((0-0)+(6-3)+(19-4)+(25-8))/4 = (0+3+15+17)/4 = 35/4 = 8.75
```

SJF:

```
@t=0, P1 selected for 4 units
@t=4, context switch 1 occurs
@t=6, P4 selected for 4 units
@t=10, context switch 2 occurs
@t=12, P3 selected for 9 units
@t=21, context switch 3 occurs
@t=23, P2 selected for 11 units
@t=34, all processes complete
Completed in 34 cycles.
Avg wait time = ((0-0)+(6-4)+(12-8)+(23-3))/4 = (0+2+4+20)/4 = 30/4 = 7.50
Avg resp time = ((0-0)+(6-4)+(12-8)+(23-3))/4 = (0+2+4+20)/4 = 30/4 = 7.50
```

RR (q=5):

```
@t=0, P1 selected for 4 units
@t=4, context switch 1 occurs
@t=6, P2 selected for 5 units
@t=11, context switch 2 occurs
@t=13, P4 selected for 4 units
@t=17, context switch 3 occurs
@t=19, P3 selected for 5 units
@t=24, context switch 4 occurs
@t=26, P2 selected for 5 units
@t=31, context switch 4 occurs
@t=33, P3 selected for 4 units
@t=37, context switch 6 occurs
@t=39, P2 selected for 1 units
@t=40, all processes complete
Completed in 40 cycles.
```

```
Avg wait time = ((0-0)+(13-4)+(33-8)+(39-3))/4 = (0+9+25+36)/4 = 70/4 = 17.50
Avg resp time = ((0-0)+(13-4)+(19-8)+(6-3))/4 = (0+9+11+3)/4 = 23/4 = 5.75
```

Note: for Avg wait time and Avg response time, you do not need to show the calculations in your output – only the final number - although it will help ensure your answer is correct if you do. Extra credit of 5 points will be awarded if you show calculations.

Provide *screenshots* or outputs showing at least 3 runs with different variables

- one of which has a context switch time of 0, two with a context switch time > 0.
- two of your runs should show at least 4 processes
- vary the quantum for 2 of your 3 processes.

Annotate the runs where appropriate.

Summarize this in a paragraph:

- What did you observe from your runs?
- Did each perform better/worse/same as you expected?
- Why or why not?

**General Guidelines**

- This project can be done as an individual assignment or as a team of two. Ensure both names are on the submission if you use a team.
- Submit all work through the dropbox in Canvas. Include your code as well (do not just provide a link to Github but you can do this as a backup if you wish).
- Pseudocode, output or screenshots, and paragraph should be provided in pdf format.
- Screen-cast-o-matic (or similar) narrated video submission showing your code running the 3 examples
- For Java submissions, submit code as a zipped project to be opened in Eclipse.
- Set up a time to meet with me via Zoom if you choose to implement in another language than Python or Java (e.g., C, C++, or C#)

**Grading**

- 70 points: code working as described above
- 10 points: screenshots /output provided of *at least* 3 runs
- 10 points: write-up of findings
- 10 points: output format is as in the example above

**Extra credit**

- 5 points: show calculations for Avg wait time and Avg response time as seen in the examples.

Below is an additional example showing a context switch time of 0.

```
Enter a seed value: 171
Enter number of processes (2, 100): 3
Enter last possible arrival time (0, 99): 9
Enter max burst time (1, 100): 10
Enter quantum size (1, 100): 4
Enter latency (0, 10): 0

3 processes created.
P     Arrival     Burst
1     2           2
2     8           8
3     8           5



Random:

@t= 2, P1 selected for 2 units
@t= 4, context switch 1 occurs
@t= 8, P2 selected for 8 units
@t=16, context switch 2 occurs
@t=16, P3 selected for 5 units
@t=21, all processes complete
Completed in 21 cycles.
Avg wait time = ((2-2)+(8-8)+(16-8))/3 = (0+0+8)/3 = 8/3 = 2.67
Avg resp time = ((2-2)+(8-8)+(16-8))/3 = (0+0+8)/3 = 8/3 = 2.67

FCFS:

@t= 2, P1 selected for 2 units
@t= 4, context switch 1 occurs
@t= 8, P2 selected for 8 units
@t=16, context switch 2 occurs
@t=16, P3 selected for 5 units
@t=21, all processes complete
Completed in 21 cycles.
Avg wait time = ((2-2)+(8-8)+(16-8))/3 = (0+0+8)/3 = 8/3 = 2.67
Avg resp time = ((2-2)+(8-8)+(16-8))/3 = (0+0+8)/3 = 8/3 = 2.67


SJF:

@t= 2, P1 selected for 2 units
@t= 4, context switch 1 occurs
@t= 8, P3 selected for 5 units
@t=13, context switch 2 occurs
@t=13, P2 selected for 8 units
@t=21, all processes complete
Completed in 21 cycles.
Avg wait time = ((2-2)+(8-8)+(13-8))/3 = (0+0+5)/3 = 5/3 = 1.67
Avg resp time = ((2-2)+(8-8)+(13-8))/3 = (0+0+5)/3 = 5/3 = 1.67



RR (q=4):

@t= 2, P1 selected for 2 units
```

```
@t=4 , context switch 1 occurs
@t= 8, P2 selected for 4 units
@t=12, context switch 2 occurs
@t=12, P3 selected for 4 units
@t=16, context switch 3 occurs
@t=16, P2 selected for 4 units
@t=20, context switch 4 occurs
@t=20, P3 selected for 1 units
@t=21, all processes complete
Completed in 21 cycles.
Avg wait time = ((2-2)+(16-8)+(20-8))/4 = (0+8+12)/3 = 20/4 = 5.00
Avg resp time = ((2-2)+(8-8)+(12-8))/4 = (0+0+4)/4 = 4/4 = 1.00
```