EOPSY LAB 3

Mohammed Nabeel Al-Mufti

301916

Warsaw/Poland

5/11/2021

TASK

Run process scheduling simulation using described below configuration. Create a configuration file in which all processes run an average of 2000 milliseconds with a standard deviation of zero, and which are blocked for input or output every 500 milliseconds. Run the simulation for 10000 milliseconds with 2 processes. Examine the two output files. Try again for 5 processes. Try again for 10 processes. Explain what's happening.

Results

Results Summary – 2 processes

```
Scheduling Type: Batch (Nonpreemptive)
Scheduling Name: First-Come First-Served
Simulation Run Time: 4000
Mean: 2000
Standard Deviation: 0
Process # CPU Time  IO Blocking CPU Completed CPU Blocked
0   2000 (ms) 500 (ms)  2000 (ms) 3 times
1   2000 (ms) 500 (ms)  2000 (ms) 3 times
```

*Processes file is attached

Despite us setting the simulation for 10000ms, we notice that it only ran for 4000ms in this simulation. The reason behind that is that each process is taking 2000ms to run, as we set it in our config file. Both the CPU time and I/O blocking are the same as what we set (500ms). The two processes had been blocked three times and registered four times one after the other in a toggling fashion – Process #0 blocked/registered > Process #1 blocked/registered -> Process #0 blocked/registered until they are both completed. Explanation behind that is as following: Process #0 is registered firstly and executes for 500ms, then it blocks. Process #1 awaits registration. Next, process #0 blocks which after I/O is open and process #1 can register. It runs for 500ms until it blocks, and the cycle continues for 3 times until processes #0 and #1 finish executing. First-come first-served principle is applied so the first registered process is the only to execute first.

Results Summary – 5 processes

```
Scheduling Type: Batch (Nonpreemptive)
Scheduling Name: First-Come First-Served
Simulation Run Time: 10000
Mean: 2000
Standard Deviation: 0
Process # CPU Time  IO Blocking CPU Completed CPU Blocked
0   2000 (ms) 500 (ms)  2000 (ms) 3 times
1   2000 (ms) 500 (ms)  2000 (ms) 3 times
2   2000 (ms) 500 (ms)  2000 (ms) 3 times
3   2000 (ms) 500 (ms)  2000 (ms) 3 times
4   2000 (ms) 500 (ms)  2000 (ms) 3 times
```

*Processes file is attached

In this case, the simulation ran for the full time that we set and each process blocked the CPU 3 times and ran for 2000 ms. After the 1st and 2nd processes were completed, We observe the registration of the 3rd and 4th process, whereas the last process is not completed. It can execute alone and does not have to wait for synchronization (Observations from both Summary-5-Processes file and Summary-5-Results file)

Results Summary – 10 processes

```
Scheduling Type: Batch (Nonpreemptive)
Scheduling Name: First-Come First-Served
Simulation Run Time: 10000
Mean: 2000
Standard Deviation: 0
Process # CPU Time  IO Blocking CPU Completed CPU Blocked
0    2000 (ms) 500 (ms)  2000 (ms) 3 times
1    2000 (ms) 500 (ms)  2000 (ms) 3 times
2    2000 (ms) 500 (ms)  2000 (ms) 3 times
3    2000 (ms) 500 (ms)  2000 (ms) 3 times
4    2000 (ms) 500 (ms)  1000 (ms) 2 times
5    2000 (ms) 500 (ms)  1000 (ms) 1 times
6    2000 (ms) 500 (ms)  0 (ms)    0 times
7    2000 (ms) 500 (ms)  0 (ms)    0 times
8    2000 (ms) 500 (ms)  0 (ms)    0 times
9    2000 (ms) 500 (ms)  0 (ms)    0 times
```

Due to the limitation of our set simulation run time (10000ms) the CPU Time of every different process was not satisfied we observe that not all the process have been ran. The first four processes have been registered, blocked and completed. The 5th and 6th processes have been registered, blocked but not completed. This is due to the fact that the following process was also blocking, and so given only 1000ms execution time and the rest didn’t have time to execute at all, and this is because of the First-Come First-Served Principle.