EOPSY Task 3: Scheduling

Piotr Kochański 289372

I declare that this piece of work which is the basis of recognition of achieving learning outcomes in the EOPSY course was completed on my own.

Piotr Kochański 289372

The goal of task 3 was to create 3 configuration files to the process scheduling simulator and observe results. The variable between the three configs was the number of processes to be scheduled - 2, 5, and 10.
Upon completing a run the simulator produced two output files containing data pertaining to each process - how much cpu time it took, how many times it blocked (Summary-Results) - as well as a detailed timeline of which process entered what state (Summary-Processes).

The only difference between the configuration files for the 2, 5, and 10 processes is the value specified to the nuprocess statement and number of ‘process’ statements. The following image shows configuration file for 10 processes, for 2 there would be 2 process 500 statements and nuprocess 2, and accordingly for 5.

```
// an average of 2000 ms
// std deviation 0
// blocked for i/o every 500 ms
// run for 10000 ms

numprocess 10

// mean deviation
meandev 2000

// standard deviation
standdev 0

// process     # I/O blocking
// every 500 ms
process 500
process 500
process 500
process 500
process 500
process 500
process 500
process 500
process 500
process 500

// duration of the simulation in milliseconds
runtime 10000
```

In the case of 2 processes both processes have more than enough time to block 3 times and complete.

```
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
```

```
Process: 0 registered... (2000 500 0 0)
Process: 0 I/O blocked... (2000 500 500 500)
Process: 1 registered... (2000 500 0 0)
Process: 1 I/O blocked... (2000 500 500 500)
Process: 0 registered... (2000 500 500 500)
Process: 0 I/O blocked... (2000 500 1000 1000)
Process: 1 registered... (2000 500 500 500)
Process: 1 I/O blocked... (2000 500 1000 1000)
Process: 0 registered... (2000 500 1000 1000)
Process: 0 I/O blocked... (2000 500 1500 1500)
Process: 1 registered... (2000 500 1000 1000)
Process: 1 I/O blocked... (2000 500 1500 1500)
Process: 0 registered... (2000 500 1500 1500)
Process: 0 completed... (2000 500 2000 2000)
Process: 1 registered... (2000 500 1500 1500)
Process: 1 completed... (2000 500 2000 2000)
```

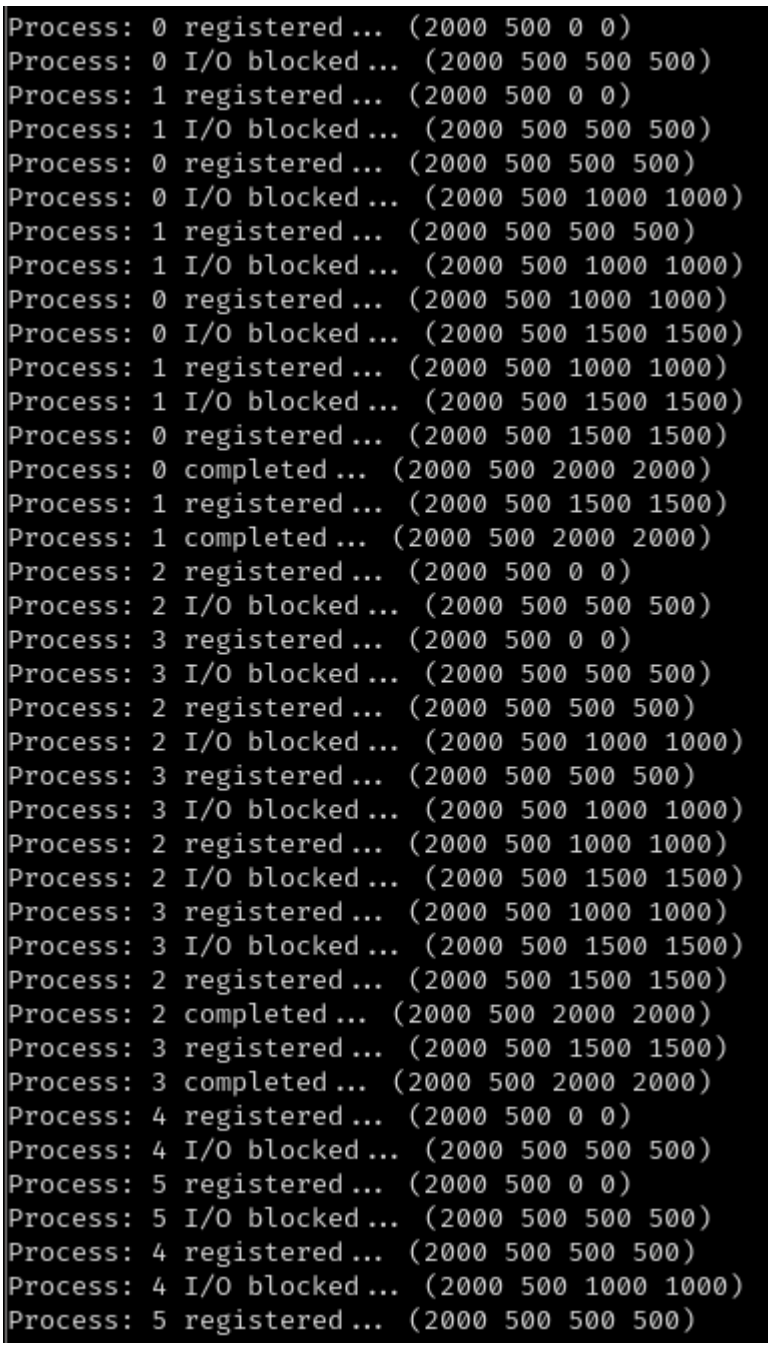
In the case of 5 processes all processes 0 through 4 show in ‘Summary-Results’ that they completed 3 times, however ‘Summary-Processes’ doesn’t show a message for process 4 completion, probably just enough time to complete but not enough to message completion.

```
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
```

```
Process: 0 registered... (2000 500 0 0)
Process: 0 I/O blocked... (2000 500 500 500)
Process: 1 registered... (2000 500 0 0)
Process: 1 I/O blocked... (2000 500 500 500)
Process: 0 registered... (2000 500 500 500)
Process: 0 I/O blocked... (2000 500 1000 1000)
Process: 1 registered... (2000 500 500 500)
Process: 1 I/O blocked... (2000 500 1000 1000)
Process: 0 registered... (2000 500 1000 1000)
Process: 0 I/O blocked... (2000 500 1500 1500)
Process: 1 registered... (2000 500 1000 1000)
Process: 1 I/O blocked... (2000 500 1500 1500)
Process: 0 registered... (2000 500 1500 1500)
Process: 0 completed... (2000 500 2000 2000)
Process: 1 registered... (2000 500 1500 1500)
Process: 1 completed... (2000 500 2000 2000)
Process: 2 registered... (2000 500 0 0)
Process: 2 I/O blocked... (2000 500 500 500)
Process: 3 registered... (2000 500 0 0)
Process: 3 I/O blocked... (2000 500 500 500)
Process: 2 registered... (2000 500 500 500)
Process: 2 I/O blocked... (2000 500 1000 1000)
Process: 3 registered... (2000 500 500 500)
Process: 3 I/O blocked... (2000 500 1000 1000)
Process: 2 registered... (2000 500 1000 1000)
Process: 2 I/O blocked... (2000 500 1500 1500)
Process: 3 registered... (2000 500 1000 1000)
Process: 3 I/O blocked... (2000 500 1500 1500)
Process: 2 registered... (2000 500 1500 1500)
Process: 2 completed... (2000 500 2000 2000)
Process: 3 registered... (2000 500 1500 1500)
Process: 3 completed... (2000 500 2000 2000)
Process: 4 registered... (2000 500 0 0)
Process: 4 I/O blocked... (2000 500 500 500)
Process: 4 registered... (2000 500 500 500)
Process: 4 I/O blocked... (2000 500 1000 1000)
Process: 4 registered... (2000 500 1000 1000)
Process: 4 I/O blocked... (2000 500 1500 1500)
Process: 4 registered... (2000 500 1500 1500)
```

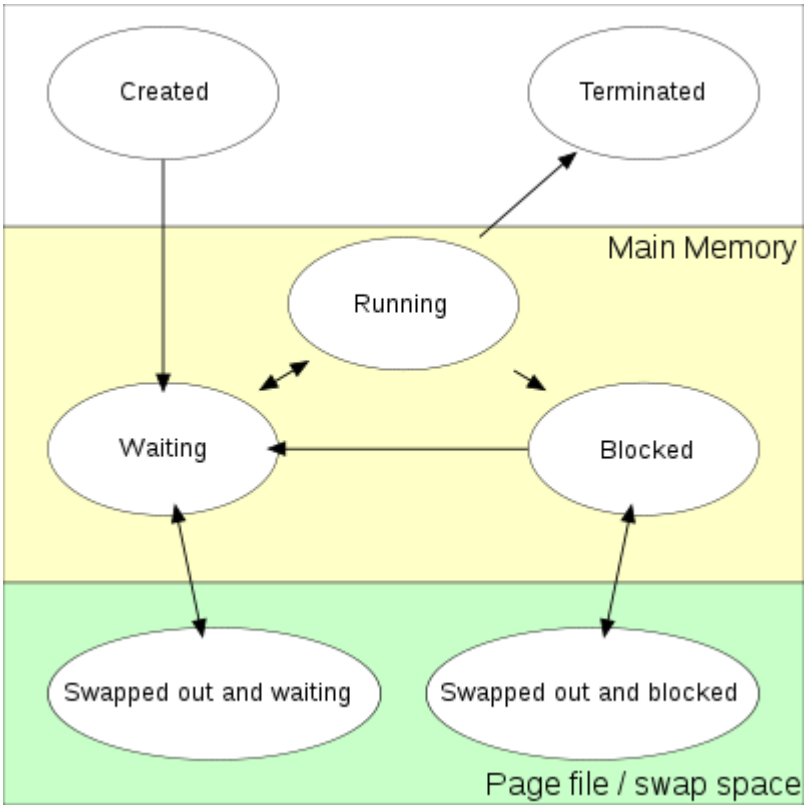
For 10 processes situation changes - there is clearly not enough time for them to all complete. There is not enough time to even create all processes.

```
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
```



We can see that processes 0-3 blocked 3 times and completed, but process 4 blocked 2 times and did not complete and process 5 blocked 1 time and did not complete either. This is because the scheduling algorithm used is first come first serve - that is the processes are run in order, when one of the blocks the i+1-th process is run.

The process states reported in the ‘Summary-Processes’ are: registered, I/O blocked and completed. These would correspond to Running, Blocked, and Terminated in the general nomenclature. Usually one would also distinguish Created and Ready states but these are not reported.



source: https://en.wikipedia.org/wiki/Process_state

When processes are first created they are in the Created state. Then when they are loaded into memory they are either executing in the Running state, waiting for some external change of state without which the process cannot continue for example some I/O or waiting for scheduler to put them in the Running state.

The scheduler is a method for awarding resources that are limited to many consumers in some specified way. In this case (process scheduler) the resource is the processor - the cpu time and the cores. The scheduler might use different algorithms based on what is to be favored - fluidity for user interaction, throughput, fairness, etc.

In the case of this simulator the scheduling method used is First Come First Serve - that is the processes are given slice of cpu time in the order they were created until they are complete. This can be checked in the ‘Summary-Results’ file in the section Scheduling Name.