EOPSY Laboratory 3

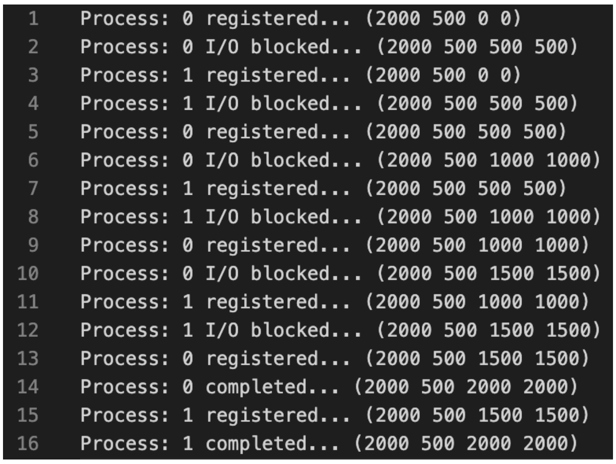
Jakub Tomkiewicz 300183

In this laboratory we are working with MOSS Scheduling Simulator, which represents the behaviour of scheduling algorithms facing a simulated mix of process loads.

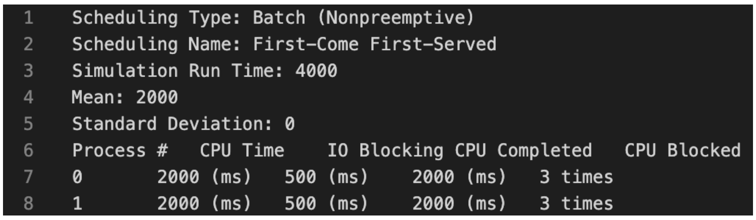
Exercise started from very simple workplace organisation and updating java with SE runtime environment in order to properly compile and run the simulator. To change parameters of simulation we must specify various parameters in the configuration file. Results of the performed simulation are visible in two files:

a) Summary-results file contains a summary report that describes the simulation and one line summary information for each process
b) Summary-processes file contains a log of actions taken by scheduling algorithm

In our task all processes run on average of 2000 milliseconds with a standard deviation of 0 and are blocked for input/output every 500 milliseconds. Simulation is running for 10000 milliseconds and with 2 processes.



```
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)
```



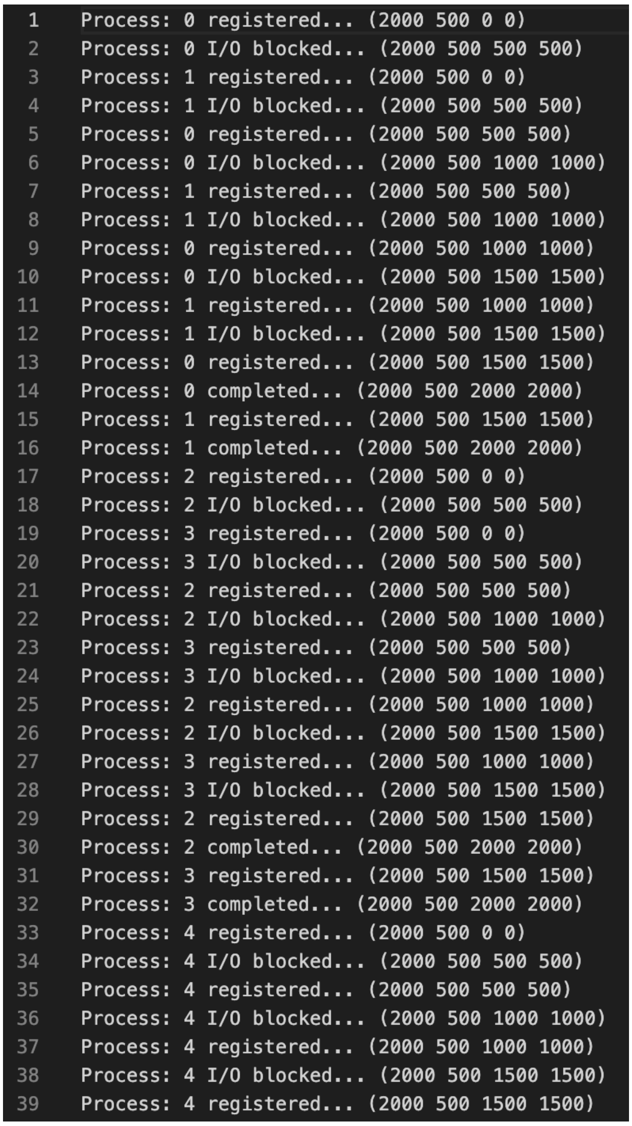
```
Scheduling Type: Batch (Nonpreemptive)
Scheduling Name: First-Come First-Served
Simulation Run Time: 4000
Mean: 2000
Standard Deviation: 0
Process #   CPU Time    IO Blocking CPU Completed   CPU Blocked
0       2000 (ms)   500 (ms)    2000 (ms)   3 times
1       2000 (ms)   500 (ms)    2000 (ms)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

The type of the scheduling algorithm is Batch (Nonpreemptive) and the name of the scheduling algorithm is First-Come First-Served. In FCFS algorithms are handled in FIFO queue, process that requests CPU first get CPU allocation first. Simulation run for 4000 milliseconds with average runtime time at 2000 milliseconds for process. Standard deviation is 0 as configured.

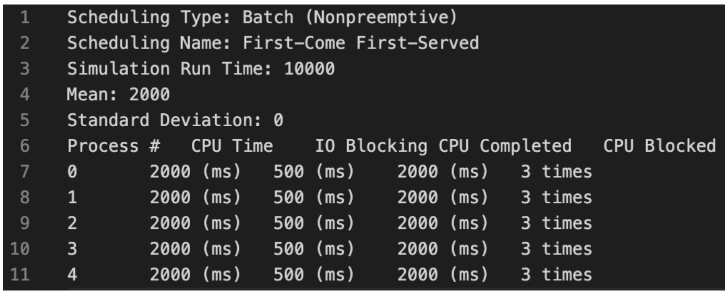
We can see 2 processes. ID starts from 0, so we are working with processes 0 and 1. For both of them total runtime was 2000 ms with input/output blocking in 500 milliseconds. Runtime for both processes is 2000 ms with both of them blocked for IO 3 times during simulation.

Log file shows alternating processes being registered and blocked to become completed at the end. Next to each process in brackets 2000 milliseconds total amount of runtime for process is located followed by 500 milliseconds time before blocking process. This two values stays the same during log for both processes. Third value in brackets is the total time that process has executed. It is increasing by 500 before registration and blockage of process. Fourth value in bracket is the same as third, according to instruction, we can remove it.

In the second part of task all the parameters stays the same except of number of processes. Now we are working with 5 processes that are blocked for input/output every 500 seconds.



```
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)
```

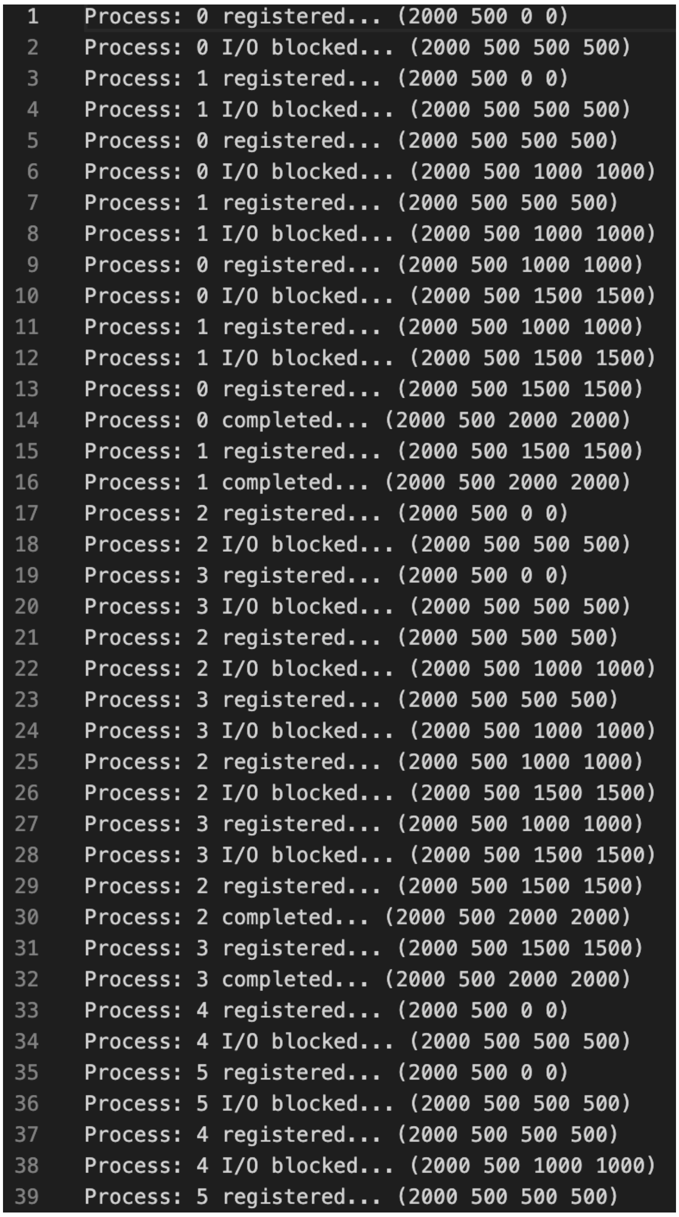


```
Scheduling Type: Batch (Nonpreemptive)
Scheduling Name: First-Come First-Served
Simulation Run Time: 10000
Mean: 2000
Standard Deviation: 0
Process #   CPU Time    IO Blocking CPU Completed   CPU Blocked
0       2000 (ms)   500 (ms)    2000 (ms)   3 times
1       2000 (ms)   500 (ms)    2000 (ms)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

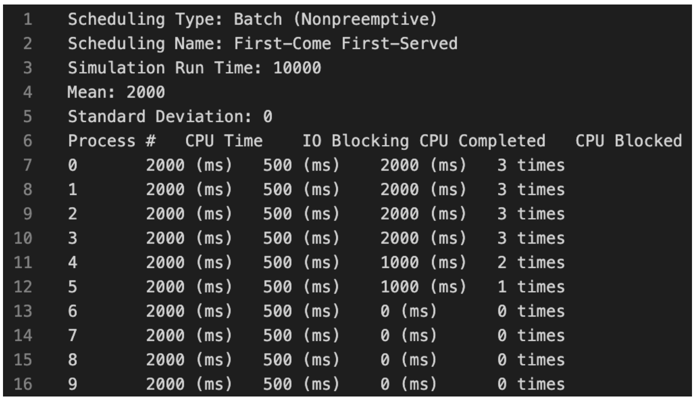
Now we can see 5 processes. All of which had total runtime of 2000 ms with IO blocking, as earlier mentioned, in 500 ms. Runtime for all processes was 2000 ms and processes were blocked 3 times during simulation.

Log file is now longer, as all 5 processes are getting into FIFO queue. At first we can see alternating processes 0 and 1 being registered and blocked until they reach 2000 ms of total execution time, then they come into completed state. After that processes 2 and 3 are registered and blocked 3 times sequentially. At the end, after processes 0, 1, 2 and 3 have completed, process 4 is being registered and blocked, this time not alternately as other processes finished. However, at this simulation we have reached simulation limit 10000 milliseconds, so at the end of the log process 4 is being registered and not completed.

Third pard of the task is the most advanced, we are working with 10 processes blocked for input/output every 500 milliseconds.



```
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: 5 registered... (2000 500 0 0)
Process: 5 I/O blocked... (2000 500 500 500)
Process: 4 registered... (2000 500 500 500)
Process: 4 I/O blocked... (2000 500 1000 1000)
Process: 5 registered... (2000 500 500 500)
```



```
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
```

Working with 10 processes is the most interesting. Everything in the summary results stays the same to and including process 3. Process 4 has now 1000 ms of execution time and has been blocked for IO 2 times, not 3 as before. Process 5 has, same as process 4, 1000 ms of execution time, but has been blocked for IO only 1 time. Processes 6, 7, 8 and 9 has 0 ms of execution time and has been blocked 0 times.

Log file is the same size as for 5 processes. It starts exactly the same for processes 0, 1, 2 and 3 that after 2000 ms of execution time and 3 IO blocks get into completed state. Afterwards process 4 is being registered and blocked 2 times alternately with process 5 registered and block only 1 time. At the end of the log process 5 is being registered, but not blocked, as we have reached simulation runtime 10000 ms. Because we have reached the set simulation time this fast only 4 out of 10 processes have reached their total amount of runtime. 2 processes has been able to reach half of their total amount of time, but 4 processes did not managed to fit in given time.