EOPSY Lab 3 - Scheduling

Author: Kacper Kamieniarz (293065)

Simulation for 2 processes

Config:

```
// # of processes
numprocess 2
// mean deviation
meandev 2000
// standard deviation
standdev 0
// time between I/O blocking (in ms)
process 500
process 500
// duration of simulation (in ms)
runtime 10000
```

First simulation involves running 2 processes with mean deviation set to 2000 milliseconds, standard deviation equal to 0 and interval between subsequent I/O blocking of 500 milliseconds.

Summary results:

```
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 results show that the type of scheduling used was Non-preemptive, specifically First-Come First-Served. In this scheduling, once the CPU time is allocated to a process, the process holds the CPU till it gets terminated or blocked (for I/O in our case). Simulation run time was 4000 ms in total with mean of 2000 milliseconds which means each of the two processes was served equally 2000 milliseconds of run time (also shown by CPU Time attribute). After 500 milliseconds of run time, each process was blocked for Input/Output and the other one of the two was executed. The processes were executed sequentially. This is why the CPU Blocked property is equal to 3 for both processes.
I took only 4000 ms for the two processes to serve 2000 ms of run-time each, therefore the simulation run time was 4000 ms and not 10000 ms as specified.

Summary process:

```
Process: 0 registered... (2000 500 0 0)
```

Process 0 status is registered, which means it is under consideration by the scheduling algorithm

```
Process: 0 I/O blocked... (2000 500 500 500)
```

After 500 ms process 0 is blocked for I/O

```
Process: 1 registered... (2000 500 0 0)
```

After process 0 is blocked, process 1 status becomes registered and it starts being executed.

```
Process: 1 I/O blocked... (2000 500 500 500)
```

Process 1 is blocked for I/O, the process in execution is process 0 again.

```
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)
```

Situation described above repeats 3 times until process 0 reaches accumulated-time of 2000 ms (last number inside of brackets) after which its status is changed to completed which means it has met or exceeded its allocated execution time (2000 ms in that case).

```
Process: 0 completed... (2000 500 2000 2000)
Process: 1 registered... (2000 500 1500 1500)
Process: 1 completed... (2000 500 2000 2000)
```

The same happens for process 1.

Simulation for 5 processes

Config:

```
// # of processes
numprocess 5
// mean deviation
meandev 2000
// standard deviation
standdev 0
// time between I/O blocking (in ms)
process 500
process 500
process 500
process 500
process 500
// duration of simulation (in ms)
runtime 10000
```

Summary results:

```
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

Results show that each one of 5 processes was served equally 2000 ms of run time, therefore the simulation run time was 10000 ms in total just as specified in the config file. The scheduling method was the same as in case of 2 processes, therefore first 4000 ms of the simulation were identical. Because the scheduling type was First-Come First-Served, the first two processes to be served 2000 ms of run time were process 0 and process 1, because as they were the first to come, they were first in the queue, before the remaining processes. The execution of processes 2 and 3 only started when process 0 and 1 have finished. Process 4 started when processes 2 and 3 have finished..

Summary process:

Process 0 and 1 start executing.

```
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 0 and 1 have completed (allocated execution time of 2000 met), processes 2 and 3 start executing.

```
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 2 and 3 complete their execution and process 4 starts

```
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)
```

Simulation for 10 processes

Config:

```
numprocess 10
meandev 2000
standdev 0
process 500
process 500
process 500
process 500
process 500
process 500
process 500
process 500
process 500
process 500
runtime 10000
```

Summary results:

```
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

Results show that 4 out of 10 processes have been given full execution time (2000 ms), two processes (4 and 5) were served 1000 ms of run-time each. Four remaining processes haven't even started executing. After the simulation time (10000 ms) has passed, processes 4 and 5 were sequentially executing. Time needed for all 10 processes to be executed in full allocated time is 20000 ms, so double the time we specified. Because the scheduling algorithm was First-Come First-Served, the processes first in the queue have finished their execution as first. Processes 4 and 5 only started executing when processes 2 and 3 have finished, the remaining processes (6 - 9) haven't even started.

Summary process:

```
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)
```

Processes 2 and 3 completed execution, processes 4 and 5 start

```
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)
```

The simulation time (10000 ms) is over, processes 4 and 5 have each executed for 1000 ms, remaining processes haven't even started at this point.