

## EOPSY Lab 3 – Scheduling

Configuration file for 2 processes:

```
1 numprocess 2
2
3 meandev 2000
4
5 standdev 0
6
7 process 500
8 process 500
9
10 runtime 10000
```

We configure 2 processes to run, blocking them after 500ms, and set all parameters according to the task requirements. Results are as follow:

---

```
1 Scheduling Type: Batch (Nonpreemptive)
2 Scheduling Name: First-Come First-Served
3 Simulation Run Time: 4000
4 Mean: 2000
5 Standard Deviation: 0
6 Process #      CPU Time      IO Blocking      CPU Completed      CPU Blocked
7 0              2000 (ms)      500 (ms)         2000 (ms)          3 times
8 1              2000 (ms)      500 (ms)         2000 (ms)          3 times
```

---

```
1 Process: 0 registered... (2000 500 0 0)
2 Process: 0 I/O blocked... (2000 500 500 500)
3 Process: 1 registered... (2000 500 0 0)
4 Process: 1 I/O blocked... (2000 500 500 500)
5 Process: 0 registered... (2000 500 500 500)
6 Process: 0 I/O blocked... (2000 500 1000 1000)
7 Process: 1 registered... (2000 500 500 500)
8 Process: 1 I/O blocked... (2000 500 1000 1000)
9 Process: 0 registered... (2000 500 1000 1000)
10 Process: 0 I/O blocked... (2000 500 1500 1500)
11 Process: 1 registered... (2000 500 1000 1000)
12 Process: 1 I/O blocked... (2000 500 1500 1500)
13 Process: 0 registered... (2000 500 1500 1500)
14 Process: 0 completed... (2000 500 2000 2000)
15 Process: 1 registered... (2000 500 1500 1500)
16 Process: 1 completed... (2000 500 2000 2000)
```

We can see two processes being run, and blocked every 500ms. Both processes get blocked 3 times, before they are completed and the simulation stops. We can see that the simulation didn't reach its full runtime – it only run for 4000ms instead of 10000. That is because there were only two processes, with 2000ms CPU time each – after they were completed, there was nothing left to simulate.

Configuration file for 5 processes:

```
1 numprocess 5
2
3 meandev 2000
4
5 standdev 0
6
7 process 500
8 process 500
9 process 500
10 process 500
11 process 500
12
13 runtime 10000
```

We change the number of processes to 5 and add blocks for these new processes. Results are as follow:

```
1 Scheduling Type: Batch (Nonpreemptive)
2 Scheduling Name: First-Come First-Served
3 Simulation Run Time: 10000
4 Mean: 2000
5 Standard Deviation: 0
6 Process #      CPU Time      IO Blocking      CPU Completed      CPU Blocked
7 0              2000 (ms)       500 (ms)         2000 (ms)          3 times
8 1              2000 (ms)       500 (ms)         2000 (ms)          3 times
9 2              2000 (ms)       500 (ms)         2000 (ms)          3 times
10 3             2000 (ms)       500 (ms)         2000 (ms)          3 times
11 4             2000 (ms)       500 (ms)         2000 (ms)          3 times
```

```
1 Process: 0 registered... (2000 500 0 0)
2 Process: 0 I/O blocked... (2000 500 500 500)
3 Process: 1 registered... (2000 500 0 0)
4 Process: 1 I/O blocked... (2000 500 500 500)
5 Process: 0 registered... (2000 500 500 500)
6 Process: 0 I/O blocked... (2000 500 1000 1000)
7 Process: 1 registered... (2000 500 500 500)
8 Process: 1 I/O blocked... (2000 500 1000 1000)
9 Process: 0 registered... (2000 500 1000 1000)
10 Process: 0 I/O blocked... (2000 500 1500 1500)
11 Process: 1 registered... (2000 500 1000 1000)
12 Process: 1 I/O blocked... (2000 500 1500 1500)
13 Process: 0 registered... (2000 500 1500 1500)
14 Process: 0 completed... (2000 500 2000 2000)
15 Process: 1 registered... (2000 500 1500 1500)
16 Process: 1 completed... (2000 500 2000 2000)
17 Process: 2 registered... (2000 500 0 0)
18 Process: 2 I/O blocked... (2000 500 500 500)
19 Process: 3 registered... (2000 500 0 0)
20 Process: 3 I/O blocked... (2000 500 500 500)
21 Process: 2 registered... (2000 500 500 500)
22 Process: 2 I/O blocked... (2000 500 1000 1000)
23 Process: 3 registered... (2000 500 500 500)
24 Process: 3 I/O blocked... (2000 500 1000 1000)
25 Process: 2 registered... (2000 500 1000 1000)
26 Process: 2 I/O blocked... (2000 500 1500 1500)
27 Process: 3 registered... (2000 500 1000 1000)
28 Process: 3 I/O blocked... (2000 500 1500 1500)
29 Process: 2 registered... (2000 500 1500 1500)
30 Process: 2 completed... (2000 500 2000 2000)
31 Process: 3 registered... (2000 500 1500 1500)
32 Process: 3 completed... (2000 500 2000 2000)
33 Process: 4 registered... (2000 500 0 0)
34 Process: 4 I/O blocked... (2000 500 500 500)
35 Process: 4 registered... (2000 500 500 500)
36 Process: 4 I/O blocked... (2000 500 1000 1000)
37 Process: 4 registered... (2000 500 1000 1000)
38 Process: 4 I/O blocked... (2000 500 1500 1500)
39 Process: 4 registered... (2000 500 1500 1500)
```

With 5 processes each taking 2000ms to run, we can see that the joint runtime is now equal to set value of 10000ms. Each process is blocked 3 times, and fully uses its time, with them all having completed 2000ms of CPU time. On the summary we can notice an interesting thing – a lack of completion message for the 4<sup>th</sup> process. This is likely due to the processor spending time on switching between processes or something similar. As a test, I changed the value of runtime in the config file to 10001ms, without changing anything else, and it was enough for the processor to complete the last process.

The order of processes being registered and processed is governed by the First-Come First-Served scheduling algorithm. This is the simplest CPU scheduling algorithm – it queues processes based on time of their arrival and prioritises those that came earlier. This means that both processes 0 and 1 need to run out their time and be fully completed, before the next two processes can get registered.

Configuration file for 10 processes:

```
1 numprocess 10
2
3 meandev 2000
4
5 standdev 0
6
7 process 500
8 process 500
9 process 500
10 process 500
11 process 500
12 process 500
13 process 500
14 process 500
15 process 500
16 process 500
17
18 runtime 10000
```

Results of the simulation:

```
1 Scheduling Type: Batch (Nonpreemptive)
2 Scheduling Name: First-Come First-Served
3 Simulation Run Time: 10000
4 Mean: 2000
5 Standard Deviation: 0
6 Process #      CPU Time      IO Blocking      CPU Completed      CPU Blocked
7 0              2000 (ms)      500 (ms)         2000 (ms)         3 times
8 1              2000 (ms)      500 (ms)         2000 (ms)         3 times
9 2              2000 (ms)      500 (ms)         2000 (ms)         3 times
10 3             2000 (ms)      500 (ms)         2000 (ms)         3 times
11 4             2000 (ms)      500 (ms)         1000 (ms)         2 times
12 5             2000 (ms)      500 (ms)         1000 (ms)         1 times
13 6             2000 (ms)      500 (ms)         0 (ms)            0 times
14 7             2000 (ms)      500 (ms)         0 (ms)            0 times
15 8             2000 (ms)      500 (ms)         0 (ms)            0 times
16 9             2000 (ms)      500 (ms)         0 (ms)            0 times
```

```

1 Process: 0 registered... (2000 500 0 0)
2 Process: 0 I/O blocked... (2000 500 500 500)
3 Process: 1 registered... (2000 500 0 0)
4 Process: 1 I/O blocked... (2000 500 500 500)
5 Process: 0 registered... (2000 500 500 500)
6 Process: 0 I/O blocked... (2000 500 1000 1000)
7 Process: 1 registered... (2000 500 500 500)
8 Process: 1 I/O blocked... (2000 500 1000 1000)
9 Process: 0 registered... (2000 500 1000 1000)
10 Process: 0 I/O blocked... (2000 500 1500 1500)
11 Process: 1 registered... (2000 500 1000 1000)
12 Process: 1 I/O blocked... (2000 500 1500 1500)
13 Process: 0 registered... (2000 500 1500 1500)
14 Process: 0 completed... (2000 500 2000 2000)
15 Process: 1 registered... (2000 500 1500 1500)
16 Process: 1 completed... (2000 500 2000 2000)
17 Process: 2 registered... (2000 500 0 0)
18 Process: 2 I/O blocked... (2000 500 500 500)
19 Process: 3 registered... (2000 500 0 0)
20 Process: 3 I/O blocked... (2000 500 500 500)
21 Process: 2 registered... (2000 500 500 500)
22 Process: 2 I/O blocked... (2000 500 1000 1000)
23 Process: 3 registered... (2000 500 500 500)
24 Process: 3 I/O blocked... (2000 500 1000 1000)
25 Process: 2 registered... (2000 500 1000 1000)
26 Process: 2 I/O blocked... (2000 500 1500 1500)
27 Process: 3 registered... (2000 500 1000 1000)
28 Process: 3 I/O blocked... (2000 500 1500 1500)
29 Process: 2 registered... (2000 500 1500 1500)
30 Process: 2 completed... (2000 500 2000 2000)
31 Process: 3 registered... (2000 500 1500 1500)
32 Process: 3 completed... (2000 500 2000 2000)
33 Process: 4 registered... (2000 500 0 0)
34 Process: 4 I/O blocked... (2000 500 500 500)
35 Process: 5 registered... (2000 500 0 0)
36 Process: 5 I/O blocked... (2000 500 500 500)
37 Process: 4 registered... (2000 500 500 500)
38 Process: 4 I/O blocked... (2000 500 1000 1000)
39 Process: 5 registered... (2000 500 500 500)

```

Simulating 10 processes with set parameters produces quite a different set of results compared to the previous simulations. The first four processes are each blocked 3 times and complete the full, 2000ms scheduled runtime. Where the difference appears are the processes 4 and 5. With the processor running the processes in pairs so to speak, process 4 is not actually completed. We can see why this happened in the processes summary. Previously the 4<sup>th</sup> process run uninterrupted on its own and now it's interjoined with process 5, just as the previous processes were with each other. This means that the processor has time to only register both processes and run each of them for 1000ms, blocking process 4 two times and process 5 only one time before it runs out of runtime. Further processes (6-9) are not even registered, because the runtime was too low for the processor to get to them.