Jan Świerczyński

293660

EOPSY

Laboratory 3*.* – *Scheduling*

Brief description of scheduling method:

Used Scheduling type: Non-Preemptive scheduling, First Come First Serve.

Non-preemptive scheduling is a CPU scheduling technique. CPU allocates its resources to the given process, the process holds CPU until it terminates or reaches specified amount of time that indicates waiting state. Processes cannot be interrupted until they are terminated or their time is up. Starvation can occur when a process with large burst time occupies the system.

First Come First Serve is a type of non-preemptive scheduling in which first process to arrive is the first process to use CPU resources. This is managed with the FIFO queue. This is the simplest form of a CPU scheduling algorithm, however not very efficient.

1st Simulation with two processes:

Configuration:

```
// # of Process
numprocess 2
// mean deviation
meandev 2000
// standard deviation
standdev 0
// process    # I/O blocking
process 500
process 500
// duration of the simulation in milliseconds
runtime 10000
```

Summary-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)
```

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
```

Conclusion:

Maximal time set for the simulation: 10 000 ms. CPU time needed for one process to be completed: 2000 ms. Due to the fact that in this particular scenario we have only 2 processes running, there was no possibility to reach the maximal simulation time.

However we can observe how processes are exactly running in our simulation and how non-preemptive scheduling looks like.

2nd simulation with 5 processes:

Configuration:

```
// # of Process

numprocess 5

// mean deviation

meandev 2000

// standard deviation

standdev 0

// process    # I/O blocking

process 500

process 500

process 500

process 500

process 500

// duration of the simulation in milliseconds

runtime 10000
```

Summary-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)
```

```
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)
```

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
```

Conclusions:

At first glance with maximal time set to 10 000 ms , with 5 processes each with a 2000 ms time of execution, we would say that it is enough to successfully terminate all of them. However, in this scenario, we can observe that only 4 processes were completed. The whole simulation ended up during the execution of last process. Here we can observe that the processes are executed in pairs and the last had to be executed alone and yet still he was blocked every 500 ms.

3rd simulation with 10 processes:

Configuration:

```
// # of Process
numprocess 10
// mean deviation
meandev 2000
// standard deviation
standdev 0
// process    # I/O blocking
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
```

Summary-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)
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)
```

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
```

Conclusions:

Similarly as for the 5 processes scenario we obtained 4 full execution during the 10 000 ms time span. 5th and 6th processes were blocked by CPU, after the 500 ms interval, respectively 2 and 1 times. Processes 7th-10th were not even registered, due to the lack of time.