

EOPSY | Lab 3

Task:

Create a configuration file in which all processes run an average of 2000 milliseconds with a standard deviation of zero, and which are blocked for input or output every 500 milliseconds. Run the simulation for 10000 milliseconds with 2 processes. Examine the two output files. Try again for 5 processes. Try again for 10 processes. Explain what's happening.

Basic description:

Scheduling Type: Batch (No-preemptive) – in this type of scheduling, when CPU allocates resources to the given process, the process holds CPU till it get terminated or reach a waiting state. Non-preemptive scheduling does not interrupt a process until they are terminated or time is up. If a process with long burst time is running CPU, then later coming process with less CPU burst time may starve

Other type of scheduling:

Preemptive scheduling is used when process change from running to ready state, or from waiting to ready state. CPU allocates resources to the given process for the limited time and then it taken away, and the process is again place back to the ready queue, if the process has still remained time. The process has a opportunity to next execution

Main differences:

In preemptive scheduling the process is allocated for limited time, on other hand in non-preemptive CPU allocates resources to the process till it terminated or switch to waiting state.

Moreover, in non-preemptive scheduling execution of process cannot be interrupted in the middle of execution. Preemptive scheduling allows interruption during execution. In addition it have a priorities of process, which can cause that low priority process has to wait a long time.

In non-preemptive scheduling the cost of CPU is significantly smaller than in preemptive

Scheduling Name: First-Come First-Served – one of type of non-preemptive scheduling algorithm, it is a simple algorithm, which queues processes in the order that they arrive. The process that comes first will be execute direct and next processes will be start after its execution. It is simple, however not very efficient.

Simulation with 2 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)
```

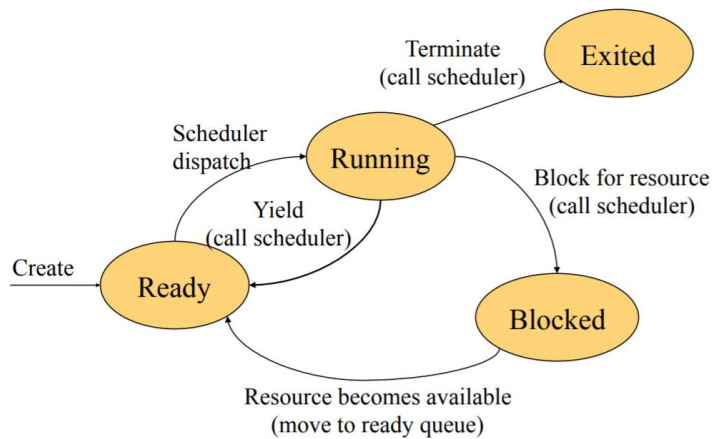
Summary-Results:

```
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

Observation:

From the configuration, we have that whole simulation time has be less or equal to 10 000 [ms]. CPU time needed to complete the one processes is 2000 [ms]. Therefore, when we are have only two processes, hence whole simulation time is 4000 [ms]. We can also observe the non-preemptive scheduling structure in example:



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

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

Observation:

First of all, we can observe that processes are executed in pairs [0&1, 2&3, 4]. From the result file, we should be able to say that 10 000 [ms] is enough to executed and terminated all 5 processes. However, we can observe, that in Summary-Processes process 4 is not completed.

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

Observation:

Similar to the case with 5 processes, we are unable to execute whole 10 processes. The first 4 processes are correctly terminated. Then, third pair was blocked by CPU three times. Moreover, the rest of processes weren't even registered.

