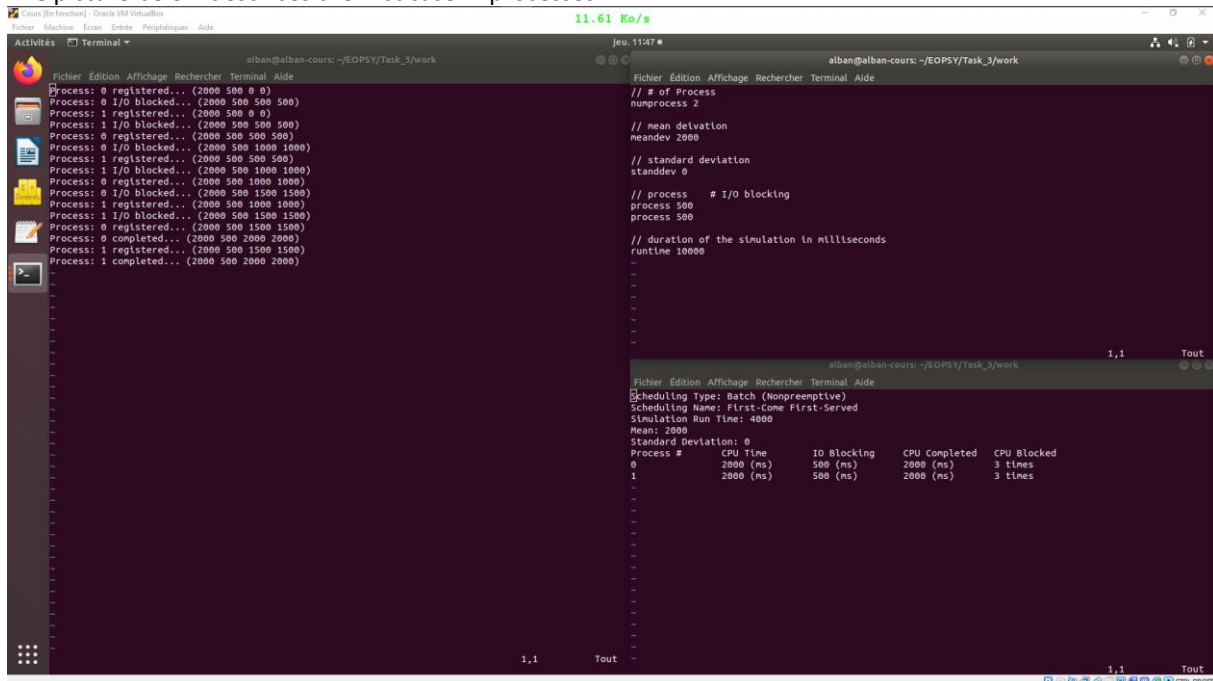


Task 3: Alban BERTHELOT

The picture below describes the first case: 2 processes.



```
alban@alban-cours: ~/EOPSY/Task_3/work
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)

// # of Process
numprocess 2

// mean delviation
meandev 2000

// standard deviation
standdev 0

// process # I/O blocking
process 500
process 500

// duration of the simulation in milliseconds
runtime 10000

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
```

We can see, on the right top corner window, the configuration file (scheduling.conf). This file defines the number of processes (2), the runtime of each process (2000 milliseconds), the standard derivation (0), the time at which the process is blocked for input or output (every 500 milliseconds) and the total time of the simulation (10000 milliseconds).

The window on the right bottom corner summarise the information put in the configuration file. This file display also how many times the process was blocked (here, both processes were blocked 3 times).

On the left window, we can see the execution order of all the processes. In this situation, we can observe that the first process starts and blocks at 500 milliseconds. Then the second process do the same thing. After that the first process start again at the previous blocking time and stop 500 milliseconds later. Such as the second process. This schema is applied until each process reach their mean time (2000 milliseconds).

The picture below describes the second case: 5 processes.

```

alban@alban-cours: ~/EOPSY/Task_3/work
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)
Process: 4 completed... (2000 500 2000 2000)

Summary-Processes 39L, 1755C
1,1 Tout

```

```

alban@alban-cours: ~/EOPSY/Task_3/work
// # of Process
nprocess 5

// mean delivation
neandev 2000

// standard deviation
standev 0

// process # I/O blocking
process 500
process 500
process 500
process 500
process 500

// duration of the simulation in milliseconds
runtime 10000

"scheduling.conf" 18L, 247C
15,11 Tout

```

```

alban@alban-cours: ~/EOPSY/Task_3/work
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

Summary-Results 11L, 397C
1,1 Tout

```

We can see, on the right top corner window, the configuration file (scheduling.conf). This file defines the number of processes (5), the runtime of each process (2000 milliseconds), the standard derivation (0), the time at which the process is blocked for input or output (every 500 milliseconds) and the total time of the simulation (10000 milliseconds).

The window on the right bottom corner summarise the information put in the configuration file. This file display also how many times the process was blocked (here, all processes were blocked 3 times).

On the left window, we can see the execution order of all the processes. In this situation we can observe that processes are run 2 by 2 (process 0 with process 1, process 2 with process 3) according to the same schema from the first case. But we can see that the last process (process 4) has run alone. It is because we have an odd number of processes and the couple of processes are made like process "odd" – process "even". Moreover, we can't see the last display of the last process because the simulation reaches it runtime in the same time that the process 4 reach is mean time.

We can deduct that the runtime (here 10000 milliseconds) is the result of the addition of mean time of each processes (every time we see "registered"). Such as :

1. Process: 0 registered... (2000 500 2000 2000) → runtime = 2000
2. Process: 1 registered... (2000 500 2000 2000) → runtime = 4000
3. Process: 2 registered... (2000 500 2000 2000) → runtime = 6000
4. Process: 3 registered... (2000 500 2000 2000) → runtime = 8000
5. Process: 4 registered... (2000 500 2000 2000) → runtime = 10000

Of course, we can divide this pattern into sub-pattern like:

- a. Process: 0 registered... (2000 500 500 500) → runtime = 500
- b. Process: 1 registered... (2000 500 500 500) → runtime = 1000
- c. ...

The picture below describes the third case: 10 processes.

```

alban@alban-cours: ~/EOPSY/Task_3/work
// # of Process
numprocess 10

// mean delvation
neandev 2000

// standard deviation
standev 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
"scheduling.conf" 23L, 308C
1,1 Tout

Fichier Edition Affichage Rechercher Terminal Aide
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
Summary-Processes 39L, 1745C
1,1 Tout
Summary-Results 16L, 508C
1,1 Tout

```

We can see, on the right top corner window, the configuration file (scheduling.conf). This file defines the number of processes (10), the runtime of each process (2000 milliseconds), the standard deviation (0), the time at which the process is blocked for input or output (every 500 milliseconds) and the total time of the simulation (10000 milliseconds).

The window on the right bottom corner summarise the information put in the configuration file. This file display also how many times the process was blocked (here, the processes 0, 1, 2, 3 were blocked 3 times, the process 4 was blocked 2 times, the process 5 was blocked one time and the other ones (process 6, 7, 8, 9) were not blocked.

On the left window, we can see the same pattern as the previous simulation. The difference between it, is that we can see the process 5. So according to my suppositions made in the previous simulation we have:

1. Process: 0 registered... (2000 500 2000 2000) → runtime = 2000
2. Process: 1 registered... (2000 500 2000 2000) → runtime = 4000
3. Process: 2 registered... (2000 500 2000 2000) → runtime = 6000
4. Process: 3 registered... (2000 500 2000 2000) → runtime = 8000
5. Process: 4 registered... (2000 500 500 500) → runtime = 8500
6. Process: 5 registered... (2000 500 500 500) → runtime = 9000
7. Process: 4 registered... (2000 500 1000 1000) → runtime = 9500
8. Process: 5 registered... (2000 500 500 500) → runtime = 10000

This explains why process 4 is blocked 2 times and process 5, only one time.

I supposed that if we increase the runtime simulation, we will see more process.