

Name:	Surname:		
NIF:	UO:		

## Exercise 1 (70% of the overall mark)

This exercise must be solved inside a directory named `UOxxxxx-Ex1`, where `UOxxxxx` is your UO.

We want to reduce the number of times that `SystemIdleProcess` is executing if there are blocked processes. With that goal, if the following conditions are met:

1. SIP have been executing for 2 or more clock interrupts (the current value of the clock interrupt counter exceeds by **at least two units** the clock interrupt number when the SIP started its execution on the last occasion).
2. Within the execution of the clock interrupt handling routine **no process has been awakened**.
3. **The first process in the sleeping queue is a user process** (IMPORTANT: nothing must be done if the first process is a daemon).

**The first process of the sleeping queue must be forced to unblock.**

Besides awakening the process, a message like the ones shown in the example must be printed, using the EXAM (x) section and magenta (@M) colour, saying that the process has been unblocked to optimize the CPU usage (SIP was using it).

With these programs:

ex1	ex2	ex3	daemonsFile
25 7 TRAP 7 10 TRAP 7 10 TRAP 3	25 7 TRAP 7 30 TRAP 7 30 TRAP 3	25 3 TRAP 7 4 TRAP 7 4 TRAP 3	ex3

And this execution (a second daemon is created from the program specified in `daemonsFile: ex3`):

```
./Simulator --debugSections=Xd --daemonsProgramsFile=daemonsFile ex1 ex2
```

This should be the resulting output:

```
[52] Process [0 - ex1] forced to wake up by OS for optimization CPU usage
[77] Process [0 - ex1] forced to wake up by OS for optimization CPU usage
[87] Process [1 - ex2] forced to wake up by OS for optimization CPU usage
[97] Process [1 - ex2] forced to wake up by OS for optimization CPU usage
[111] The system will shut down now...
[113] END of the simulation
```

And with this other execution (`ex3` becomes a user process):

```
./Simulator --debugSections=Xd ex1 ex2 ex3
```

This should be the resulting output:

```
[37] Process [2 - ex3] forced to wake up by OS for optimization CPU usage
[47] Process [0 - ex1] forced to wake up by OS for optimization CPU usage
[57] Process [0 - ex1] forced to wake up by OS for optimization CPU usage
[67] Process [1 - ex2] forced to wake up by OS for optimization CPU usage
[77] Process [1 - ex2] forced to wake up by OS for optimization CPU usage
[91] The system will shut down now...
[93] END of the simulation
```



## Exercise 2 (30% of the overall mark)

This exercise must be solved inside a directory named `UOxxxxx-Ex2`, where `UOxxxxx` is your `UO`.

We want to compute, for each process, the number of times that it moves from EXECUTING to READY, separating the cause of the transition. The solution must distinguish whether the transition was caused because the process was thrown out of the processor or because of a system call.

When a process terminates, no matter why, messages like the ones in the example must be printed, using the EXAM (x) section and magenta (@M) colour.

With these programs:

ex4	ex5	ex6	ex7
30	30	30	30
22	22	6	6
ADD 5 5	ADD -5 -5	ADD -5 0	ADD -3 0
TRAP 4	TRAP 4	TRAP 4	TRAP 4
INC -1	INC 1	TRAP 7	INC 1
ZJUMP 2	ZJUMP 2	INC 1	ZJUMP 2
JUMP -3	JUMP -3	ZJUMP 2	JUMP -3
DIV 3 0	TRAP 3	JUMP -3	TRAP 7 120
		TRAP 7 80	HALT
		IRET	

And this execution (that changes the default maximum number of programs from 4 to 8 and, therefore, reduces the amount of memory available for each process to a half):

```
./Simulator --debugSections=X --numProcesses=8 ex4 ex5 ex6 ex7
```

This should be the resulting output:

```
[497] Process [1 - ex5] was thrown out of the processor [1] times
[497] Process [1 - ex5] transferred the control of the processor [10] times
[531] Process [0 - ex4] was thrown out of the processor [3] times
[531] Process [0 - ex4] transferred the control of the processor [9] times
[581] Process [2 - ex6] was thrown out of the processor [0] times
[581] Process [2 - ex6] transferred the control of the processor [1] times
[696] Process [3 - ex7] was thrown out of the processor [0] times
[696] Process [3 - ex7] transferred the control of the processor [2] times
[706] Process [7 - SystemIdleProcess] was thrown out of the processor [2] times
[706] Process [7 - SystemIdleProcess] transferred the control of the processor [0] times
```