Lab Session 5: Final Review
24296 - Sistemes Operatius

The objective of this lab session is to do a review of all subjects that we explained in the course. We provided a list of short programs numbered from 1 to 7. You will need to fix or complete them and write the answer with what you added/changed in a document. You will need to deliver the pdf document via Aula Global Monday 15 March before 12h. Always compile the exercise file and execute it before. Here the exercises:

- **01-Processes-ExecCmds.c** The program executes right now the command "ps" but we want to execute all the commands of the array "cmds". What we are missing in the code?
- **02-Processes-Sequential.c** The program creates 10 processes and runs them concurrently. You will need to execute them sequentially. The printing of the index "i" must then be always in order.
- **04-Threads-NoThreads.c** The program creates 10 threads and executes them sequentially. Remember the sequential schema makes no sense for threads. Change the code, calling directly the thread function, doing the same, without creating any thread.
- **04-Threads-Join.c** Same code than previous exercise. Now change it code to execute the threads concurrently and fix any problems with the printing of the thread id (related to parameter passing),
- **05-Synch-Sum.c** The provided code creates threads that sum in different positions of the array "sums" and is correctly synchronized with a global lock. Make it more efficient with a lock per each position of the array. Write the lines of code that you changed in the document.
- **06-Comm-Pipe.c** The program creates 10 processes and runs them concurrently. Each process uses the index "i" as a process identifier. Do the same thing but passing the identifier with a pipe. The father will pass the id as an int the child that will write it.
- **07-Comm-FileSem.c** The provided code creates in the beginning a binary file sums.dat that contains 4 zero integers. Then creates 10 processes that synchronize using a mutex named semaphore the access to this file adding 1 in every position. Make it more efficient with a named semaphore per each position of the array. Explain what you changed in the document.