Seminar 5: Final Review
24296 - Sistemes Operatius

The objective of this seminar is to do a review of the course with the following exercises that (except exercise 2) are related to the last lab session that you did.

1. Write a program that will execute the following array of commands concurrently:

   ```
   char* cmds[][5] = {{"ps", NULL}, {"ls", "-la", NULL}, {"cat", "data.txt", NULL}};
   ```

2. The shell command below does the following: (a) lists the files in the current folder (b) writes them to a pipe which (c) is read by shell command "wc" that counts the number of lines with the option "-l". Implement it using system calls fork, execvp, pipe and dup2.

   ```
   > ls -la | wc -l
   ```

3. Write a program that creates 10 threads and executes them concurrently as it is usual for threads and wait for them to finish. Now discuss why it makes no sense to execute them sequentially and change the code, calling directly the thread function, doing the same, without creating any thread.

4. Write a program that creates 10 threads and pass to each of them the following data that each will wtite to screen: (a) an integer identifier (b) two integers (c) a char pointer string.

5. Write a program that is going to do a collaborative sum creating 10 threads into an integer array of 2 positions. Each thread with an odd id will do 10 sums into position 0 of the array and each thread with even id will do 10 sum into position 1. Synchronize the code correctly and make the code more efficient with a lock per each position of the array.

6. Write a program that creates 10 processes and runs them concurrently. Each process receives its identifier (an integer) through a pipe and writes it to the screen.

7. Write a program that creates a binary file with 2 integers initialized to 0. Then write the same program than exercise 5 with processes instead of threads and using named semaphores as synchronization.

Delivery exercises

We are going to program a chain of data exchange in two different ways:. We are going to do it with an array using monitor synchronization and with pipes.

1. Create a program that will create a chain of 10 Threads which will exchange a number in the following way: Each thread will be waiting on a position of an array to be signaled when it has been filled with the number. When this happens, it will copy the number to the following array position for the next thread to get it. Synchronize the threads with monitors as simply as possible.

2. Create a program that will generate a chain of 10 processes and each two consecutive processes will be communicated with a pipe, forming a chain structure in which each process reads from a pipe and writes to the following one. Then make the father introduce a number to the first process and gather the result of the chain. Each process will read, increment and pass to the following process that number.