In this exercise we familiarize ourselves to threads.

Exercise 12 (Thread synchronization, 1p)

Phase 1. (Demonstrate the problem)

Write a program that has a main thread and 10 sub-threads. Each sub-thread increments the common counter as fast as possible 1000 000 times (if there is no problem, let them increment 10 000 0000 times). The threads also increment their own counter in the array in the memory area of the main thread (counter array is a local variable in the main function).

The main thread waits until all sub threads have done their work. After that the main thread displays the value of the counter, the list that tells how many times each thread has incremented their own counters and also the value how many times the increment of common counter has been missed. Observe how many increments have been lost. What is the reason that some of the increments have been missed?

Phase 2. (Fix the problem)

Fix the problem so that no increments are lost any more. It is not allowed to use any delays in the increment loop or any busy loops for waiting.

Note: If you want to use threads in your program, compile the program with the option -lpthreads.

Exercise 12 (Extra, socket server using threads, 0.25p)

Change the implementation method of the server you developed in exercise 10, so that you now use threads instead of processes to guarantee the concurrent service of all clients.