

## HW2 – Threads.

In this task you should train data structures and threading skills.

The background for this task is this:

You have a endless datastream that should be processed fast. Say a security camera data, where an intrusion should be detected.

In our assinent we will use a random number generator (provided) that simulates the endless stream, and your task is to count the number of prime numbers in this stream.

The idea is to parallelize the process, so all the cpu core are utilized.

You may improve the isPrime function too, if nessesary.

Note that you allowed to use **only 2MB of ram** space.

You can implement the solution in any language you want but:

a) beat my **best** results (on your pc) and explain why this happens:

b) be able to explore the cpu and memory consumption of your solution in details (**exceptc c/c++**). your finding should be presented in class (in other words, show us how you can **deeply** explore the structure of a language you have used, not just a debugger)

The task in details:

You are given a random number generator called randomGenerator (provided).

You also get a basic implementation of the prime counter app, called primeCounter (provided).

The usage is like this:

randomGenerator <seed> <num\_of numbers>.

F.ex randomGenerator 10 10 will give you ten numbers, generated with random seed “10”.

primeCounter don’t need any arguments, but it supposed to get the numbers with std in.

So the usage with the random generator looks like this:

```
./randomGenerator 10 100 | ./primesCounter
```

the expected output is : “5 total primes.”

the amount of numbers that likely to be tested is between 1 to 1000 **milions** of numbers, so performance is the name of the game.

You can use the “time” command to check the time it takes to calc the primes.

What to submit:

a) your best solution code, and a way to run it (instructions, makefile, etc.)

b) printScreen with 10M numbers checked by provided primeCounter (used for reference) + time

c) printScreen with 10M numbers checked by your solution + time

d) proof of ram usage below 2MB

expected results (or closed to it):

- you should do a least 4 times better without improving the isPrime function (90% mark)
- you should do a least 10 times better with improving the isPrime function (10% mark)
- the best solution in class will be invited to lunch by the tutor

**WARNING: changing cpu to performance mode may overheat your machine. If you are not sure about it, use a PC at the campus labs. No warranties is provided to your own device WARNING**

Keep in mind that there may be automatic test for this excersize, so don't change the output format

Good luck !

Hint: a lock free queue is explained here:

<https://www.schneems.com/2017/06/28/how-to-write-a-lock-free-queue/>

Technical Notes:

- Submission day – up to 21/02/24 08:00 am
- Put your results in ZIP (not rar/7z/tar etc.) and name it by your ID. (-15 points if not)
- You can sibmit in pairs, name id ID\_ID (-15 points if not)
- Failing to submit working Make may lower your point dramatically, down to Zero !
- Be aware of linux/windows files transfer. Linux is CaseSensitive, while Windows is not
- Don't wait for the last minute, as “My VM is died”, “My cat swallowed the mouse” and other subterfuges will not be accepted