System Programming
4th Laboratory (22 and 24 March 2017)

I

The **lab4_1.c** program prints the amount of multiples of 7 and 19 betwee 0 to UINT_MAX.
Compile the program and execute the following command:

time ./lab4_1

Take note of the time it takes do compute the values.
Using the **time** man page verify what each value means.

II

Since current computers have more than one core, it is possible to take advantage of them to parallelize this code.
Create three child processes and each one one third of the numbers to be verified.
After the conclusion, each process should print the partial computed values.
Use the proposes data structure (in a array) to transfer the limits from the parent to each child.

0 UINT_MAX/3	UINT_MAX/3+1 UINT_MAX*2/3	UINT_MAX*2/3+1 UINT_MAX
P1	P2	P2

III

Change the previous program so that the parent prints the total number of multiples after the conclusion of all children.
Change the communication data structure to store the computed results.
Create a shared memory region before doing the fork. The parent will write the limits before the fork and read the partial results after the wait.
Use the following functions:

- **`exit`**
- **`wait`**
- **`mmap`**

IV

Implement a system composed of three programs:

- **gen_random** that generates random positive integers and writes them sequentially in a shared memory region. This program will terminate after 10 seconds printing how many odd and even numbers were generated.
- **count_even** – a program that reads the shared memory and counts how many even numbers were written by gen_random
- **count_odd** – a program that reads the shared memory and counts how many odd numbers were written by gen_random

The three processes are not related and should be launched from command line (not by fork).
Can you guarantee that all generated number are counted by **count_odd** or **count_even**?

References

https://www.cs.cf.ac.uk/Dave/C/node27.html
http://menehune.opt.wfu.edu/Kokua/More_SGI/007-2478-008/sgi_html/ch03.html
http://www.ibm.com/developerworks/aix/library/au-spunix_sharedmemory/
http://www.kohala.com/start/unpv22e/unpv22e.chap12.pdf