System Programming
4th Laboratory (20, 21 and 23 March 2018)

I

Implement a system that counts how many prime numbers there are in a sequence of integers.
This system is composed of a master process (that generates a sequence of random numbers and writes those numbers on a pipe) and slaves (that continuously read integers from a pipe and verify if those numbers are prime).
The master receives as arguments (argv), the number of slaves and the length of sequence of values.
The master process will generate random numbers (between 0 and 99999) in a sequential manner and write them to pipe to be processed by one slave.
Each slave, after the processing of all values each slave should print the number of prime values calculated. These processes do not know how many numbers will be generated.

NOTE: every process reading from a pipe is notified when that pipe is last closed for writing (i.e. no more processes are writing to it).

II

Modify the previous programs so that the child processes do not print the count of prime values, but send to the parent its partial count. Only the parent prints the sum on the screen.

III

Modify the previous exercise so that each slave writes in a second pipe every prime number found (after being processed).
This new pipe should be read by the master, that will print those prime numbers on the screen.

IV

What happens if the amount of numbers generated in the master is so large so that the pipe gets full before the slaves process them and the master can not write anything more? Modify the solution to guaranteed that it works even in this condition.

V

Using FIFOs implement a system composed of several producers (that reads text lines from the keyboard and writes to predetermined FIFO) and a single consumers that reads those lines and prints them on the screen in upper case.
These processes are not related and all receive the name of the FIFO as an argument (argv).

V

Using FIFOs implement a system composed of a producer (that reads text lines from the keyboard and writes them to a predetermined FIFO) and a set of consumers (each line is rread by a single consumer that prints it on the screen in upper case.
These processes are not related and all receive the name of the FIFO as an argument (argv).

REFERENCES

http://tldp.org/LDP/lpg/node7.html
http://beej.us/guide/bgipc/output/html/multipage/index.html