TP 09 : Input/Output Redirection and Pipes

1 Review

In order to run shell commands on C, we use the `execlp` command. See file "exec.c" to see how this function works.

— Add the following line after the `execlp` command.

```
printf("Execlp successfully executed!\n");
```

What do you observe ? Why does it happen ? Check the manual to see if you can figure out.

— Use `fork` in order to correct this code. Hint : you will also need to use `wait`.

— Now, finally, write into the file `"pingReader.txt"` the input from the `execlp`. Hint : Use the `dup2` command you saw in class.

2 Shady files

Move the "obsf" file into an empty directory and run it. Two (empty) files should have appeared in your current directory. Delete them using only the command line.

Note : how are files identified in general ?

3 Wrong program

Consider `closed_pipe.c`. The son is supposed to print the characters sent to him by the parent. Explain the error you see, and correct the program.

4 Pipes and code replacement

1. Last week, we used exit signals to communicate between parent and child processes. Modify the code for `simple.c` from last week to use pipes instead.
2. Write a program (in C) that downloads the archive `bootstrap.zip` (which can be found at `https://lmf.cnrs.fr/downloads/Margulies/bootstrap.zip`), and unzips it without creating a temporary file. In other words, we want the command `curl <url> | unzip` in C. The curl and unzip programs will be called by `execlp`.

5 Buffered and Unbuffered output

Look at the file `buffered.c`. What do you expect the output to be when you have a look ? What is the actual output. Can you explain the discrepancy ?

6 The function of Hénon

We will calculate the orbit of a dynamic system of dimension 2. The function of Hénon is described by the system

$$H_{a,b} = \begin{cases} x_{n+1} = a - by_n - x_n^2 \\ y_{n+1} = x_n. \end{cases}$$

We will use one process to calculate the sequence $(x_n)_n$ and another process to calculate the sequence $(y_n)_n$.The processes will exchange their data via one (or more) pipe (s).

Subsequently, we will also create a process dedicated to the exit : this process must write lines in the form `0.3415 1.2451` where the first number is x_n and the second y_n in a file `henon.dat`.

We can plot the function with the command `gnuplot henon.p` after having downloaded the script "henon.p". The file `henon.dat` must be in the same folder as `henon.p`.

Observe the graph you get for values $a = 1.4$ et $b = -0.3$.