TP7

The course homepage is here:

`http://www.lsv.ens-cachan.fr/~schwoon/enseignement/systemes/ws1415/`.

You will find the slides from the course and some other files for the exercise there.
Details of shell commands and C functions can be obtained by using the `man` command.

1 Fork

Consider the programs `fork.c` on the webpage.

1. Without running the program, what do you expect to be its output?
2. If we draw a graph illustrating the child processes spawned by each proces, what is the shape of that graph?
3. Download and compile the program `forktree.c` (to be made available during the course; it is a variant of pstree showing only processes with "fork" in their name) and use it to visualize the tree from (b). For this, make each process wait for ten seconds before terminating, then run `forktree` in a different window.
4. Now make each process wait a random amount of time, between 10 and 60 seconds, before terminating (see the `drand48` manpage). Regard the development of the tree using `forktree`. What do you observe?

2 Calculator

On the course homepage you will find a little calculator for expressions with natural numbers, addition, multiplication, and subtraction.
Useful man pages: fork(2), wait(2) – see WEXITSTATUS for the latter.
Use `make` to compile. You only need to modify `main.c`.
Change the program so that instead of computing everything directly, it invokes two child processes *in parallel*, which compute the values of the left and right subexpression, respectively, and return the result through their exit status. The parent waits for the two children, then computes the result.
It is easier to implement this for multiplication and addition than for subtraction. Why?

3 Programming a shell

The object of this project is to program a simple bash-like shell that will execute programs, redirect their output etc. Naturally, the use of the system(3) function is forbidden...

On the project homepage you will find the code skeleton of a simple shell. It contains a simple command-line interpreter that understands output redirection and command chaining (pipes, sequences etc). But currently, it only knows how to implement the very simplest of commands. We will complete it little by little.

For the beginning, make the shell execute sequences of commands chained by sequences, AND, and OR.