Process and Fork

1 Exercises

For all the exercices you will use these following functions:

- fork(2)
- wait(2)
- waitpid(2)
- exec(3)

You must include the following headers :

```
1  #include <sys/types.h>
2  #include <sys/wait.h>
3  #include <unistd.h>
```

1.1 Simple fork

For this exercise, you just have to write un program that create 10 child process. Each process should print his PID 10 time.

- if the fork fail, you must exit the program with 1.
- Otherwise it must return 0.

Reminder : In your head, in your head they are crying
In your head, in your head
Zombie, zombie, zombie-ie-ie

1.2 Nano-42SH

During the "magnifique séminaire C/UNIX organisé par les ASM" you discovered how to read from the standard input "stdin".
Now you will have to write your own "shell". Read the instruction from stdin and execute it with exec(3) function.

- if the fork fail, you must exit the program with 1.
- Otherwise it must return 0.
- if the exec return an error. You must handle it and return the same error code.

1.3 Pipe me

Now, it is time to learn a little bit more about file descriptors and processes !
First of all, read man 2 pipe !!!

Reminder : When forking a child process, all file descriptors are duplicated.

Now, you will have to write a program which takes exactly one parameter (argv[1]). It will create a pipe and fork the process :

- The child process will write the content of the parameter in one of the file descriptor (returned by the pipe) and exit with the last return value of the last write **You can't write more that 256 char in one time**.
- The parent process will read in the other file descriptor, print the result on the standard output and return the exit status of the child process.

You will stop your program and return -1 if there is no parameter and -2 if a syscall failed. Otherwise, your program must return 0.
Be carefull : your program needs to work with very very long strings as parameter ! Don't forget about zombies !
To write on stdout (standard output), you can use puts(3) or printf(3) (man 3 puts, man 3 printf).

We strongly advise you to have a look at : man 3 exit, man 2 fork, man 2 write, man 2 read, man 2 waitpid.