UFAZ - Bachelor of Computer Science

System Programming

PW09 : pipes

For each exercice, we expect the student to write a program, compile it and run it without errors of several examples. Test sets and comments are as important as the code itself.

Exercice 1

Write a program made of 2 processes: the parent reads data from the standard input and transmit them to its child through a pipe, the child then prints them on the standard output. The parent then waits for the termination of the child process.

To start with, we suggest that you write a function

```
void copy(int fdsrc, int fddst)
```

which copies the contains of a file whose descriptor is `fdsrc` to a file whose descriptor is `fddst` and then use this function in a simple (single-process) program to copy the standard input into the standard output. You may test your function with:

```
$ ./a.out < /bin/ls > toto
$ cmp /bin/ls toto
```

If `cmp` does not display any error, then the copy went fine.

Exercice 2

Write a program which create 2 child processes, redirect I/O and call the function `execlp` to do the same as the shell command : `ps eaux | grep "^<name>" | wc -l`.

The name shall be given by the user as an argument of the program, otherwise it is the value of the environment variable `USER` by default.

Exercice 3

Generalize exercice 1 to n processes: the first one passes data from the standard input to the second process, which then passes them on to the third and so on until the $(n-1)^{th}$ which passes them on to the n^{th} process (the parent) which displays it on the standard output. There must be $n-1$ pipes and the $n-1$ children must be directly linked to the parent process which executes the `main` function.