{EPITECH LEARN DIFFERENT*}

MINISHELL2

THIS ISN’T FLYING. THIS IS FALLING WITH STYLE!

MINISHELL2

Preliminaries

binary name: mysh
language: C
groupe size: 1
compilation: via Makefile, including re, clean and fclean rules
authorized functions: malloc, free, exit, opendir, readdir, closedir, getcwd, chdir, fork, stat, lstat, fstat, open, close, getline, strtok, strtok_r, read, write, execve, access, isatty, wait, waitpid, wait3, wait4, signal, kill, getpid, strerror, perror, strsignal, *pipe*, *dup*, *dup2*

- ✓ The totality of your source files, except all useless files (binary, temp files, objfiles,...), must be included in your delivery.
- ✓ All the bonus files (including a potential specific Makefile) should be in a directory named bonus.
- ✓ Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

Project

The goal of the project is to enrich your *minishell1* project by adding semicolons (‘;’), pipes (‘|’), and the four redirections (‘>’, ‘<’, ‘>>’, ‘<<’) management.

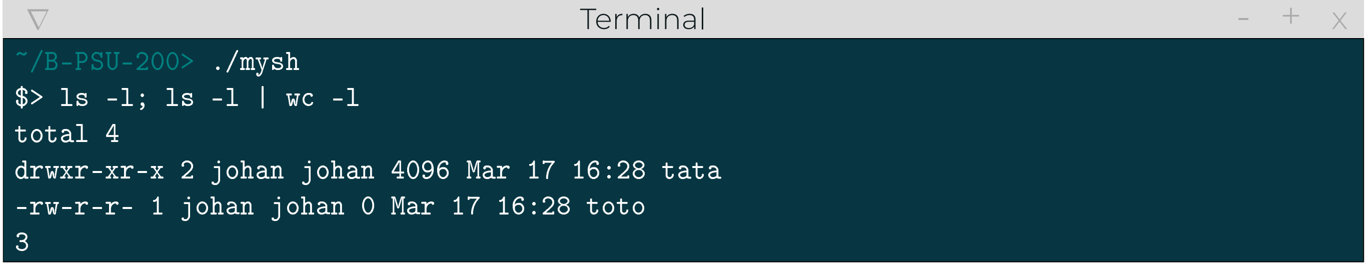
The priority management is a key part of this project.

> To start, make sure to have a functionnal minishell1 as you are going to need execute binaries.

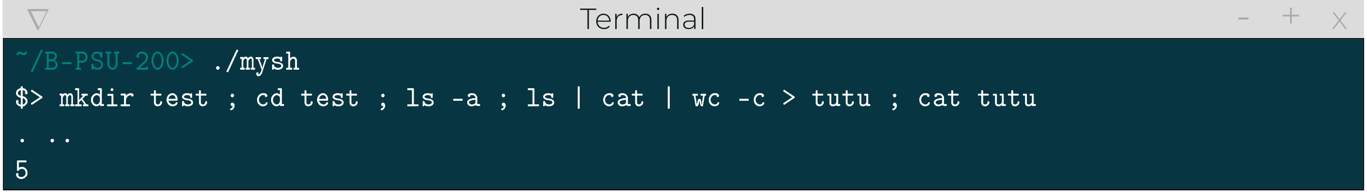
Error output, standard input and standard output redirections will be considered as bonuses. Such as 2>&1

Examples

You can use any prompt as we are testing your minishell as followed :
echo “command” | ./mysh



```
~/B-PSU-200> ./mysh
$> ls -l; ls -l | wc -l
total 4
drwxr-xr-x 2 johan johan 4096 Mar 17 16:28 tata
-rw-r-r- 1 johan johan 0 Mar 17 16:28 toto
3
```



```
~/B-PSU-200> ./mysh
$> mkdir test ; cd test ; ls -a ; ls | cat | wc -c > tutu ; cat tutu
. ..
5
```

v1

{EPITECH}

LEARN DIFFERENT*