DESCRIPTION

This project is an implementation of a linux shell in C, with basic commands. To compile it, juste type "make" in the command line of another shell (lets call it S), then to run it, write "./shell" in the command line of S.

QUESTIONS

QUESTION 1

execvp() is preferred over other exec functions because it returns an error if the specified program is not found or if there is an error executing the program, making it easier to handle these cases.

QUESTION 2

The symbol for the sequence operator in bash is ";". An example of a command where the sequence operator behaves differently from the and operator is:

>>> cd dossier [; / &&] cat readme

"&&" est paresseux et va s'arrêter si dossier n'est pas trouvé, alors que ";" va tout de même executer cat readme.

QUESTION 3

skip

QUESTION 4

The role of parentheses in the command (cmd1 && cmd2 | cmd3 ...) 2>/dev/null is to group the commands together and to specify the order in which the commands should be executed. The parentheses are used to indicate that the commands inside the parentheses should be executed before the redirection 2>/dev/null.

An example of a command that uses these parentheses non-trivially is

>>> (cmd1 && cmd2) | cmd3 >>> cmd1 && (cmd2 | cmd3)

The parentheses are used to group together multiple commands and specify the order in which they should be executed, as well as to override the default precedence of the logical operators (in this case, && and |).

QUESTION 5

When you type CTRL+C in the shell, it sends the SIGINT signal to the current process, which usually terminates the process. To recover after typing CTRL+C, you can catch the SIGINT signal and define a custom behavior for it. For example, you can ignore the signal by calling the signal function and setting the signal handler to SIG_IGN, or you can define a custom signal handler function that performs certain actions before exiting.

QUESTION 6

When you enter the command "ls > dump", the shell redirects the standard output of the "ls" command to the file "dump" rather than displaying it on the terminal.

QUESTION 7

Just using dup2 doesn't work as it only works with file descriptors, not with arbitrary processes. For example :

```
>>> cat file1 | sort | uniq > file2
```

Here, simply using dup2 doesn't work because the pipe needs to connect to the output of one command to the input of another one, not the file descriptor of a file to the input of a command.

EXAMPLES

```
>>> pwd

>>> ls

>>> pwd [&&/||/;] cd file1 (with or without file1 exiting)

>>> cd file1 [&&/||/;] pwd

>>> mkdir file1

>>> cd file1

>>> cd ..

>>> rmdir file1

>>> ls | wc

>>> to be continued
```

BONUS

I implemented the ls, cat and cd commands.

-ls and cat : execvp already handles ls and cat, as well as many other commands (nano, touch, mkdir, ...).

-cd : I made a special case in the C_PLAIN case.