Promotion 2023

Année A4

Rapport Operating Systems Project

Souhail AIT LAHCEN

**Rapport Operating Systems Project
Année 4**

12/2021 - Document confidentiel

Table des matières

Report Operating Systems Project 2021-2022

Souhail AIT LAHCEN

31 Décembre 2021

I Introduction

For this project, I need to create a interactive shell.
We can fin my script, report etc.. in my repository GitHub :
`https://github.com/Souhail99/Project-Operating-Systems`.

Also, **as you can see in my README.md on my repository GitHub**, you don't need
(in your terminal) to write **/usr/bin/python** before the name of my code but you need to
put the all path , because I use this :

```
1    #!/usr/bin/env python
2
```

II Basic Shell

First, our script (shell) need to basic, If I run my script I have this :

```
mysh$
```

For exit, I need to write **exit** :

```
mysh$ exit
guignole@souhail-ubuntu:~/Documents/Projet OS$
```

III Built-in Commands

In this project, we need to implement built-in command like **cd, pwd, and exit**. First, I
define some variable like c :

```
result=""
c=True
```

This variable is useful for exit command, while the user don't write exit the script run :

1

```
while c:
    #In this part, I try to test 2 command like local and exit (to stop the program)
    code = input("mysh$ ")
    if code=='exit' or (code.startswith("cd ") or code == "cd"):
        if code =='exit':
            c=False
        elif code.strip() == "cd":
            os.chdir(os.environ["HOME"])
        else:
            if len(code.split(" "))==2:
                os.chdir(code.split(" ")[1])
            else:
                newcode=''
                c=code.split(" ")
                for i in range(1,len(c)):
                    if i == 1:
                        newcode+=c[i]+" "
                    else:
                        newcode+=c[i]
                os.chdir(newcode)
    #In this part, I recover the user's command and I split it. Then, I find in my computer the list of path where we can find the commands, then to fi
    #In the others cases
    else:
        commande(code)
#endregion
```

   As you can see, the while loop depend of c.
So for **cd** is more complex, we need to unsterstand three cases, as you can see in my
script, if the command is only cd I return in the HOME file like this :

```
mysh$ pwd
/home/guignole/Documents/Projet OS
mysh$ cd
mysh$ pwd
/home/guignole
mysh$
```

And I return in the previous directory like this :

```
/home/guignole
mysh$ cd Documents
mysh$ pwd
/home/guignole/Documents
mysh$ cd Projet OS
mysh$ pwd
/home/guignole/Documents/Projet OS
mysh$
```

So the second or the third case is for the case if the user put several direcotory like in the
previous pictures : **cd Documents and cd Projet OS**.

IV   Redirection

   We need to have the possibility to write some redirections, a simple example is **>**, so I
resolve this with that :

```
# Let's imagine that we have the case of cat text.txt>text2.txt so without a space so we need to add a spac
if ">" in code and " " not in code:
    listes.extend(code.split(">"))
    listes.extend(" ")
    listes[len(listes)-1]=listes[len(listes)-2]
    listes[len(listes)-2]=">"
```
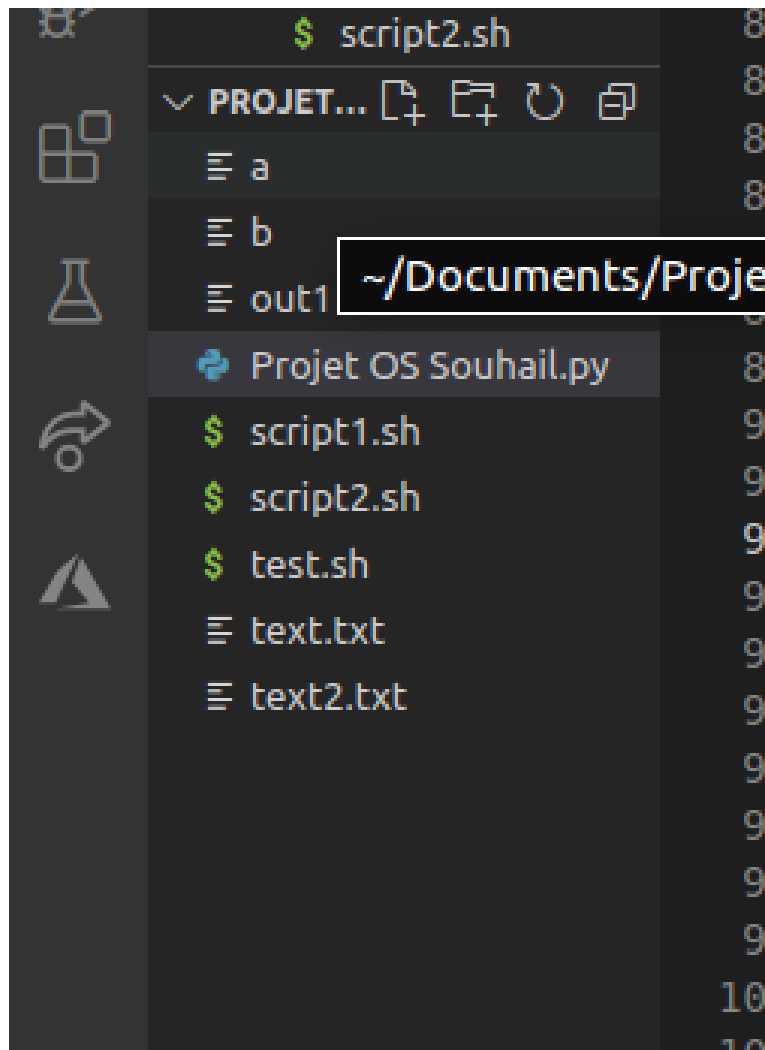
And an example is this :
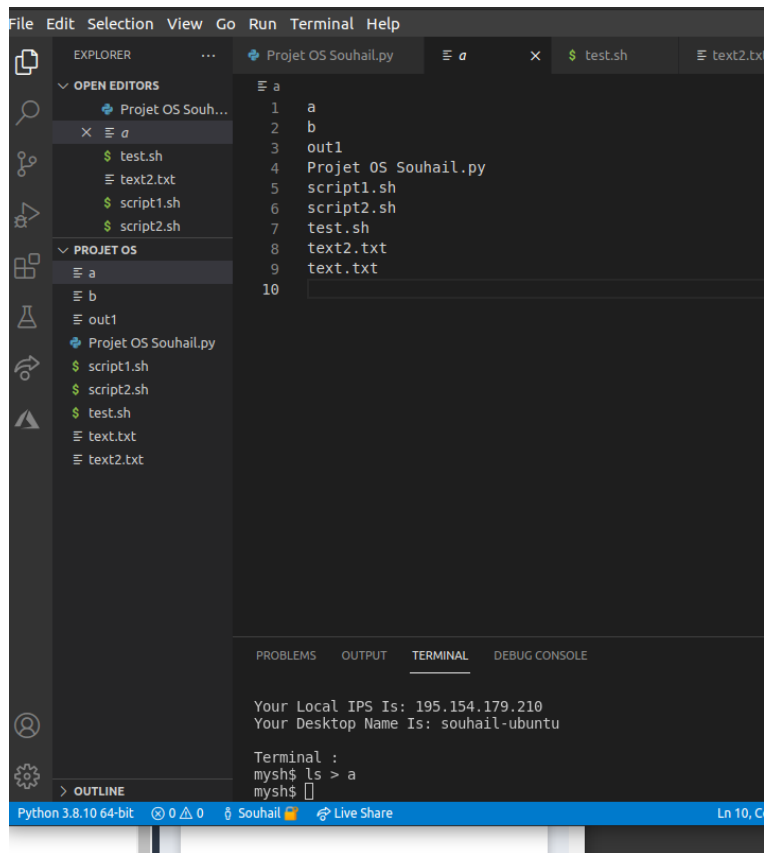
```
Terminal :
mysh$ ls > a
mysh$
```
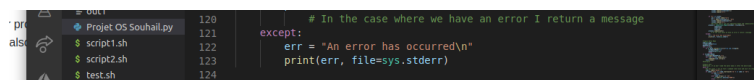
2

a is in the list of files :

$ script2.sh

∨ PROJET...

☰ a

☰ b

☰ out1 ~/Documents/Proje

🐍 Projet OS Souhail.py

$ script1.sh

$ script2.sh

$ test.sh

☰ text.txt

☰ text2.txt

Finally, it contains this :

V Program Errors

We need to return an error with this :
print(err, file=sys.stderr)
so the code is :


```
                    # In the case where we have an error I return a message
120
121    except:
122        err = "An error has occurred\n"
123        print(err, file=sys.stderr)
124
```
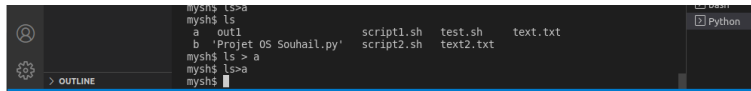
An example :


```
mysh$ cdd
An error has occurred

mysh$
```
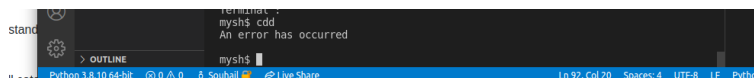
VI   White Spaces

So yes, even if, we write a command we need to permit the user to write command like :
mysh$ ls
mysh$ ls > a
mysh$ ls>a
So, as we can see it works :

Finally, the three commands work.

VII   Batch Mode

To conclude, I want to speak about the **Batch Mode** : I test the batch mode with this file :

Then, with this code :

I test it in the linux terminal, so I have this :

It works !

VIII   Test Case

For this part, let's imagine we want to use a script sh like that :

Script 1:

```
# script1.sh
echoerr() { echo "$@" 1>&2; }
echoerr hello world
```

So, this is work in my script because I have this :