



Promotion 2023

Année A4

Rapport Operating Systems Project

Souhail AIT LAHCEN

**Rapport Operating Systems Project
Année 4**



Table des matières

I	Introduction	1
II	Basic Shell	1
III	Built-in Commands	1
IV	Redirection	2
V	Program Errors	4
VI	White Spaces	5
VII	Batch Mode	5
VIII	Test Case	6

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 find my script, report etc.. in my repository GitHub :
<https://github.com/Souhail99/Project-Operating-Systems>.

Also, you don't need (in your terminal) to write **/usr/bin/python** before the name of my code, because I use this :

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

This first line is for the user want to use the project without of need to write (/bin/ python etc...).

II Basic Shell

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

```
Terminal - 1
mysh$
```

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

```
Terminal - 1
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 :

```

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 = ""
                ccode = code.split(" ")
                for i in range(1, len(ccode)):
                    if i == 1:
                        newcode += ccode[i] + " "
                    else:
                        newcode += ccode[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 find
    #In the others cases
    else:
        commande(code)
    c = True

```

As you can see, the while loop depends on `c`.
 So for `cd` is more complex, we need to understand 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 directories 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 space
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] = ">"

```

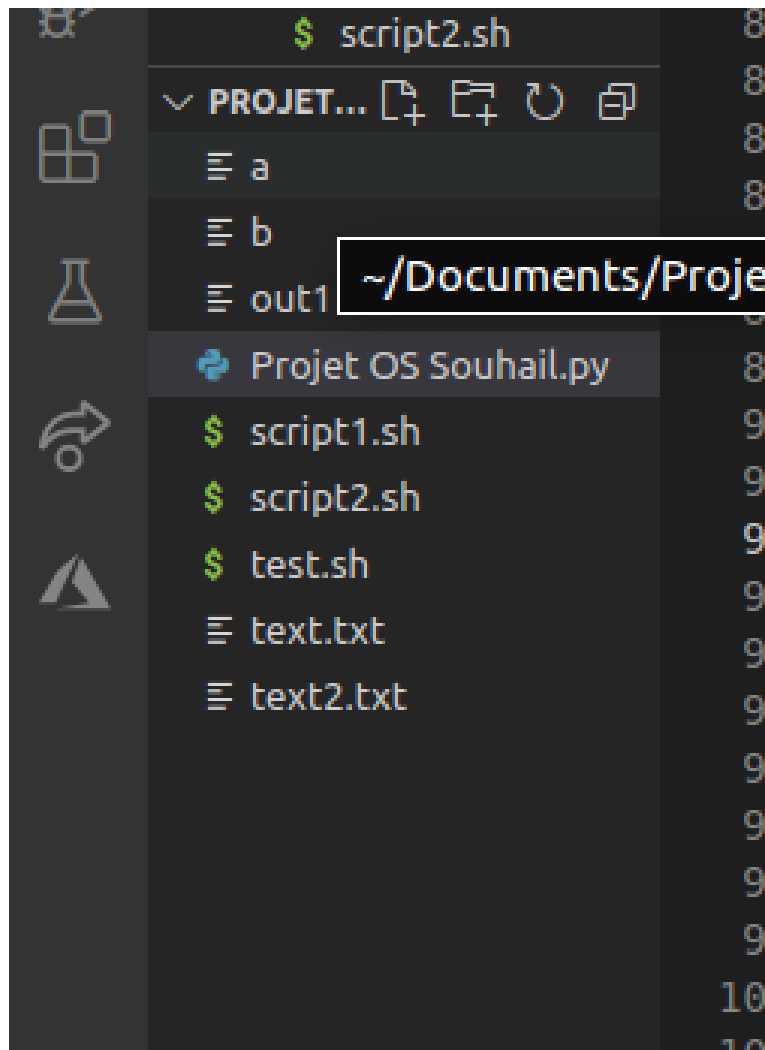
And an example is this :

```

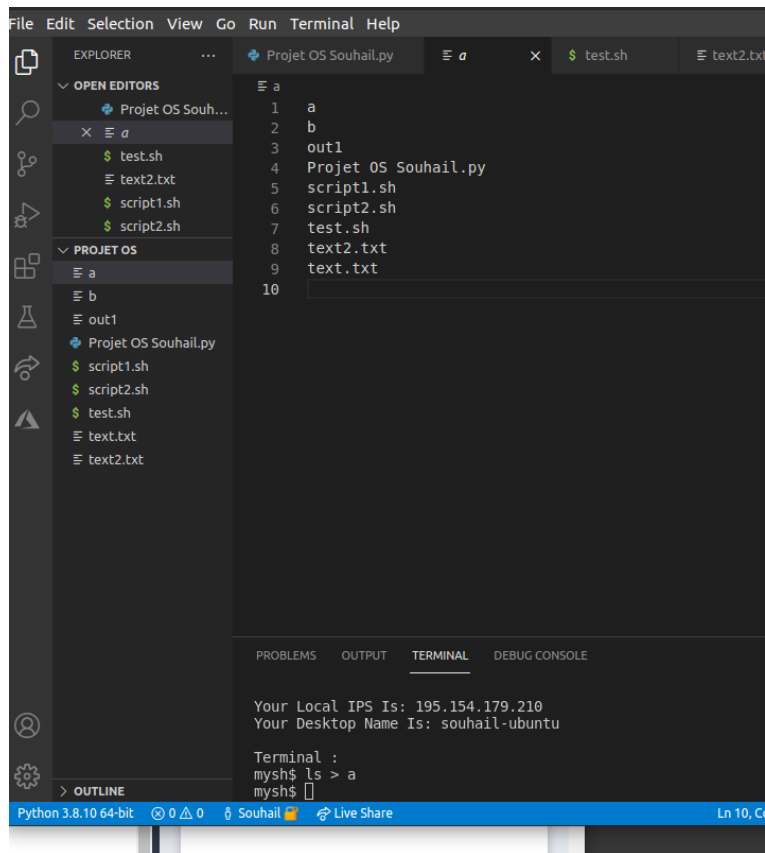
> OUTLINE
terminal:
mysh$ ls > a
mysh$

```

a is in the list of files :

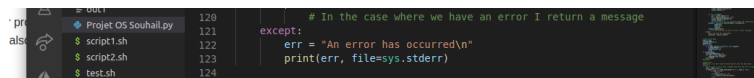


Finally, it contains this :

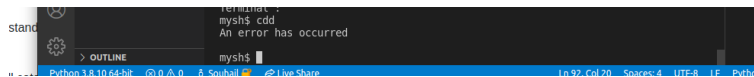


V Program Errors

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



An example :



VI White Spaces

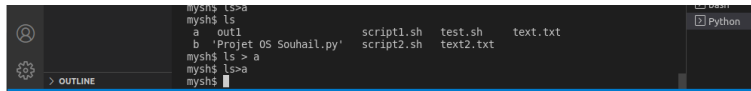
So yes, even if, we write a command we need to permit the user to write command like :

```
mysh$ ls
```

```
mysh$ ls > a
```

```
mysh$ ls>a
```

So, as we can see it works :

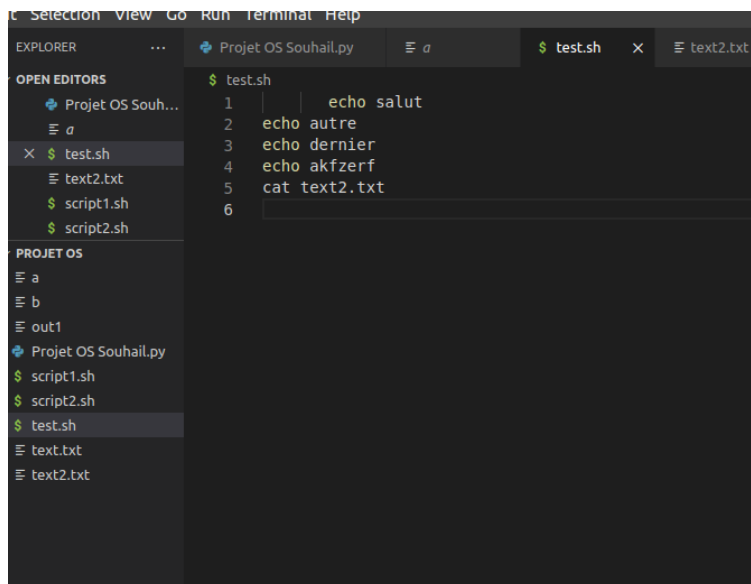


```
mysh$ ls>a
mysh$ ls
a out1
b 'Projet OS Souhail.py' script1.sh test.sh text.txt
mysh$ ls > a
mysh$ ls>a
mysh$
```

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 :



```
Selection View Go Run Terminal Help
EXPLORER ...
OPEN EDITORS
  Proj OS Souh...
  a
  test.sh
  text2.txt
  script1.sh
  script2.sh
PROJET OS
  a
  b
  out1
  Proj OS Souhail.py
  script1.sh
  script2.sh
  test.sh
  text.txt
  text2.txt

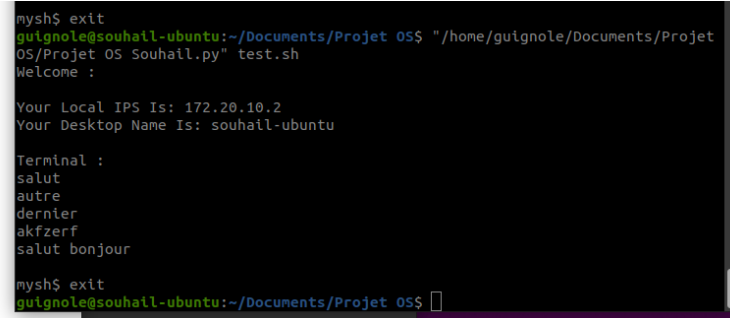
$ test.sh
1 echo salut
2 echo autre
3 echo dernier
4 echo akfzerf
5 cat text2.txt
6
```

Then, with this code :



```
124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
$ script1.sh
$ script2.sh
#region Batch Mode
arg=sys.argv
fileBatchMode=False
NameOfFile=""
for i in arg:
    if os.path.isfile(str(i))==True and i!=arg[0]:
        fileBatchMode=True
        NameOfFile=str(i)
        break
if fileBatchMode:
    file=open(NameOfFile,"r")
    for line in file.readlines():
        if "exit" not in line.rstrip():
            commande(line.rstrip())
        else:
            c=False
    file.close()
#endregion
```

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



```
mysh$ exit
guignole@souhail-ubuntu:~/Documents/Projet OS$ "/home/guignole/Documents/Projet OS/Projet OS Souhail.py" test.sh
Welcome :

Your Local IPS Is: 172.20.10.2
Your Desktop Name Is: souhail-ubuntu

Terminal :
salut
autre
dernier
akfzerf
salut bonjour

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

It works !

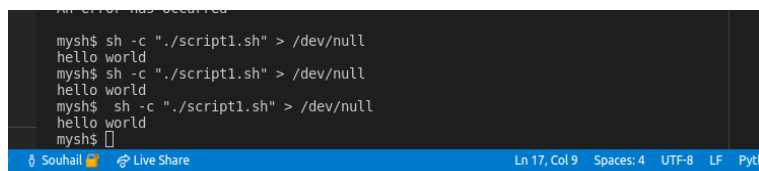
VIII Test Case

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 :



```
mysh$ sh -c "./script1.sh" > /dev/null
hello world
mysh$ sh -c "./script1.sh" > /dev/null
hello world
mysh$ sh -c "./script1.sh" > /dev/null
hello world
mysh$
```