Andrew Phakeovilay 1B

Programmation système

Projet 2

Table des matières

A. La commande cat > fichier....2
1. Script....2
2. Tests....2
B. La commande ls -l | grep \\.py....3
1. Script....3
2. Tests....4

Implémentations de commandes unix avec redirections

A. La commande cat > fichier

1. Script

```python
import os, sys, signal

if len(sys.argv)!=2:
    if(len(sys.argv)>2):
        print("Nombre d'arguments invalide\n")
        os._exit(1)
    print("Nom du fichier ne peut pas être vide\n")
    os._exit(1)


def handler(numero, frame):
    os._exit(0)

try:
    signal.signal(signal.SIGINT, handler)
    fd = os.open(sys.argv[1], os.O_WRONLY | os.O_CREAT | os.O_TRUNC, 0o644)
    entree = os.read(0,8)
    while len(entree)>=0:
        os.write(fd, entree)
        entree = os.read(0, 8)

except OSError as e:
    print(e.strerror)
    os._exit(1)
```

2. Tests

On exécute le fichier fonctionnellement en passant en argument le nom du fichier, puis nous pouvons écrire dans le fichier jusqu'à qu'on quitte par CTRL-C. On lit le nouveau fichier crée avec la commande cat.

```
[0]aphakeovilay@barbara:~$ python3 Projet2_script1.py fichier
Hello World!
^C[0]aphakeovilay@barbara:~$ cat fichier
Hello World!
```

Les cas d'erreurs sont qu'il n'y a pas de nom de fichier ou il y a trop d'arguments..

```
[0]aphakeovilay@barbara:~$ python3 Projet2_script1.py
Nom du fichier ne peut pas être vide

[1]aphakeovilay@barbara:~$ python3 Projet2_script1.py a a
Nombre d'arguments invalide
```

B. La commande ls -l | grep \\.py

1. Script

```python
import os

try:
    (r,w) = os.pipe()
    pid = os.fork()
    if pid:
      if os.fork():
        os.close(r)
        os.close(w)
        os.wait()
        os.wait()
        os._exit(0)
      else:
        os.close(w)
        os.dup2(r,0)
        os.read(os.execlp("grep", "grep", "\\.py"), 64)
    else:
        os.close(r)
        os.dup2(w, 1)
        os.execlp("ls", "ls", "-l")
except OSError as e:
    print(e.strerror)
    os._exit(1)
```

2. Tests

```
[1]aphakeovilay@barbara:~$ python3 Projet2_script2.py
-rw-r--r-- 1 aphakeovilay etudiants2a   632 nov.  13 20:40 myshell.py
-rw-r--r-- 1 aphakeovilay etudiants2a    94 oct.  11 10:33 prog1.py
-rw-r--r-- 1 aphakeovilay etudiants2a   292 oct.  11 10:34 prog2.py
-rw-r--r-- 1 aphakeovilay etudiants2a   106 oct.  11 11:47 prog3.py
-rw-r--r-- 1 aphakeovilay etudiants2a   574 déc.   4 18:15 Projet2_script1.py
-rw-r--r-- 1 aphakeovilay etudiants2a   463 déc.   4 18:16 Projet2_script2.py
-rw-r--r-- 1 aphakeovilay etudiants2a    82 oct.  21 09:56 tp4.py
-rw-r--r-- 1 aphakeovilay etudiants2a   312 oct.  28 17:03 TP5.py
[0]aphakeovilay@barbara:~$ ls -l
total 160
-rw------- 1 aphakeovilay etudiants2a    13 déc.   4 18:12 fichier
drwxr-xr-x 2 aphakeovilay etudiants2a  4096 nov.  13 20:38 miniprojet1
-rw-r--r-- 1 aphakeovilay etudiants2a   632 nov.  13 20:40 myshell.py
-rw-r--r-- 1 aphakeovilay etudiants2a    94 oct.  11 10:33 prog1.py
-rw-r--r-- 1 aphakeovilay etudiants2a   292 oct.  11 10:34 prog2.py
-rw-r--r-- 1 aphakeovilay etudiants2a   106 oct.  11 11:47 prog3.py
-rw-r--r-- 1 aphakeovilay etudiants2a 89930 oct.  11 11:26 ProgSys_TP01_CR_PHAKEOVI
LAY_Andrew_1B.odt
-rw-r--r-- 1 aphakeovilay etudiants2a 30240 oct.  11 12:12 ProgSys_TP02_CR_PHAKEOVI
LAY_Andrew_1B.odt
-rw-r--r-- 1 aphakeovilay etudiants2a   574 déc.   4 18:15 Projet2_script1.py
-rw-r--r-- 1 aphakeovilay etudiants2a   463 déc.   4 18:16 Projet2_script2.py
prw------- 1 aphakeovilay etudiants2a     0 oct.  28 15:48 tmp
-rw-r--r-- 1 aphakeovilay etudiants2a    82 oct.  21 09:56 tp4.py
-rw-r--r-- 1 aphakeovilay etudiants2a   312 oct.  28 17:03 TP5.py
[0]aphakeovilay@barbara:~$
```