
COMPTE RENDU PROJET PARTIE 2

Christopher Marie-Angélique – Groupe 2B

Table des matières

Partie 2 : Implémentation de commande unix avec redirection (> ou)	2
A. La commande cat > fichier	2
Code source :	2
Explication :	2
B. La commande ls -l grep \\.py	3
Code source :	3
Explication :	3

Partie 2 : Implémentation de commande unix avec redirection (> ou |)

A. La commande cat > fichier

Dans cette exercice, il nous ait demandé de créer un programme python qui reproduit le fonctionnement de la commande **cat > fichier** qui permet d'écrire dans un fichier

Code source :

```
from os import *
import sys

try:
    fic = open(sys.argv[1], O_WRONLY | O_CREAT | O_TRUNC, 0o644)

    buffer = read(0, 255)

    while len(buffer):
        write(fic, buffer)
        buffer = read(0, 255)

    close(fic)

except OSError as e:
    traceback.print_exc()
    print(e.strerror)
    exit(1)
```

Explication :

Dans un premier temps nous ouvrons le fichier placé en argument. Dans le cas où il n'existe pas, nous le créons.

Ensuite, nous lisons la sortie standard grâce à la méthode « read ».

Dans une boucle, nous écrivons dans le fichier ouvert nous lisons la sortie standard.

Lorsque la boucle est terminée, nous fermons le fichier.

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

Dans cet exercice il nous a été demandé de créer un programme python qui reproduit le fonctionnement de la commande `ls -l | grep \\.py` qui permet de filtrer les résultats en ne gardant que les lignes contenant `.py`.

Code source :

```
from os import *
import sys

try:
    (rd, wd) = pipe()

    PID = fork()

    if PID:
        PIDfork()
        2 = f
        if PID2:
            close(wd)
            close(rd)
            wait()
            wait()
            sys.exit(0)
        else:
            dup2(wd, 1)
            close(rd)
            execlp("ls", "ls", "-l")
    else:
        dup2(rd, 0)
        close(wd)
        execlp("grep", "grep", "\\\.py")

except OSError as e:
    traceback.print_exc()
    print(e.strerror)
    exit(1)
```

Explication :

Dans un premier temps nous créons une pipe.

Dans une condition, nous vérifions si nous sommes dans le processus père, si c'est le cas, nous le dupliquons pour créer un fils grâce à la méthode « fork ».

Si nous sommes dans le processus fils, nous fermons l'écriture et la lecture avec la méthode « close ».

Par la suite, nous exécutons la commande « ls -l » pour lister la liste des fichiers et des répertoires grâce à la méthode « exectp ».

Ensuite nous combinons à cette commande, la commande « grep \\.py » de la même manière pour ne lister que les fichiers contenant « .py ».