



Introduction aux systèmes d'exploitation

D.TOUAZI FAYCAL

MAITRE DE CONFÉRENCES

UNIVERSITÉ M'HAMED BOUGARA – BOUMERDES

f.touazi@univ-boumerdes.dz

A photograph of a modern staircase made of grey concrete or metal treads. The stairs are illuminated from below by small, glowing green lights, creating a step-like pattern of light and shadow. The perspective leads up towards the top right of the frame.

Groupement de commandes, pipes/tubes, redirections

f.touazi@univ-boumerdes.dz

L'exécution des commandes

Groupement de commandes, pipes/tubes, redirections ...

Chaque commande une fois exécutée renvoie un code de retour qui sera dans la variable **\$?** :

- 0** si tout s'est bien passé
- 1** si il n'y a pas de résultat
- 2** si il y a une erreur

```
user1@PC:~$ grep -n user3 /etc/group
68:user2:x:1002:user1,user3
69:user3:x:1003:user2
user1@PC:~$ echo $?
0
user1@PC:~$ grep -n user5 /etc/group
user1@PC:~$ echo $?
1
user1@PC:~$ grep -n user3 /etc/grou
grep: /etc/grou: Aucun fichier ou dossier de ce type
user1@PC:~$ echo $?
2
```

Groupement de commandes

Le point virgule (;

Le point virgule ; est utilisé pour exécuter plusieurs commandes sur la même ligne

commande1 ; commande2; ...

Exemple :

```
user1@PC:~$ whoami; cd Bureau; hostname; pwd
user1
PC
/home/user1/Bureau
user1@PC:~/Bureau$
```

Le pipe ou tube ()

Le pipe ou tube (|) permet de rediriger directement la sortie d'une commande vers d'entrée d'une autre commande

commande1

| **commande2**

...

```
user1 @ PC:~$ cat fichier
1 z
2 y
3 x
4 w
user1 @ PC:~$ cat fichier | sort -k 2
4 w
3 x
2 y
1 z
user1 @ PC:~$ cat fichier | sort -k 2 | grep -n w
1 : 4 w
```

Les caractères (&&)

commande1

&& commande2

commande2 sera exécutée uniquement si **commande1** a
retourné
0 (réussite)

Exemple :

```
user1@PC:~$ ls Bureau/  
TP1 TP2  
Bureau/TP1  
Bureau/TP2  
user1@PC:~$ rm -f Bureau/*  
user1@PC:~$
```

Les caractères (||)

commande1 || commande2

commande2 ne sera exécutée que si
commande1 a retourné un code différent de **0**

Exemple :

```
user1@PC:~$ grep user1 /etc/passwd | echo "utilisateur inexistant"
user1:x:1001:1001:,:/home/user1:/bin/bash
user1@PC:~$ grep user10 /etc/passwd || echo "utilisateur inexistant"
utilisateur inexistant
```

Les canaux

On peut considérer un canal comme un fichier, qui possède son propre descripteur par défaut, et dans lequel on peut ou lire ou écrire

Le canal d'entrée standard se nomme `stdin` et porte le descripteur **0 (exemple clavier)**

Le canal de sortie standard se nomme `stdout` et porte le descripteur **1 (exemple l'écran)**

(exemple d'erreur standard se nomme `stderr` et porte le descripteur l'écran)

```
user1@PC:~$ pwd ; HOSTNAME  
/home/user1
```

```
HOSTNAME: commande introuvable  
user1@PC:~$
```

stderr

Redirection le canal de sortie standard vers un fichier

Les caractères **>, >> ou (1>, 1>>)** permettent rediriger la sortie standard (**stdout**) vers un fichier et le crée s'il n'existe pas

>: contenu de fichier sera écrasé par les nouvelles données (stdout)

```
user1@PC:~$ pwd > fichier
user1@PC:~$ cat fichier
/home/user1
user1@PC:~$ hostname > fichier
user1@PC:~$ cat fichier
PC
```

>>: Les nouvelles données (stdout) seront inséré à la fine du

fichier

```
user1@PC:~$ cat fichier
PC
user1@PC:~$ whoami >> fichier
user1@PC:~$ cat fichier
PC
user1
```

Redirection le canal d'erreur standard vers un fichier

Les caractères **2>** , **2>>** permettent rediriger le canal d'erreur (**stderr**) vers un fichier et le crée s'il n'existe pas

2> : contenu de fichier sera écrasé par les nouvelles données (stderr)

```
user1@PC:~$ { pwd ; HOSTNAME; } 2> fichier  
/home/user1  
user1@PC:~$ cat fichier  
HOSTNAME: commande introuvable
```

2>>: Les nouvelles données (stderr) seront inséré à la fine du

fichier

```
user1@PC:~$ cat fichier  
PC  
user1@PC:~$ { pwd ; HOSTNAME; } 2>> fichier  
/home/user1  
user1@PC:~$ cat fichier  
PC  
HOSTNAME: commande introuvable
```

Redirection les deux canaux vers un fichier

Les caractères **2>&1** à la fine de la commande permettent rediriger les deux canaux (**stdout et stderr**) vers un seul et même fichier et le crée s'il n'existe pas.

commande > fichier 2>&1

contenu de fichier sera écrasé par les nouvelles données (**stdout et stderr**)
user1@PC:~\$ pwd **HOSTNAME** > fichier 2>&1
user1@PC:~\$ cat fichier
/home/user1
HOSTNAME: commande introuvable

commande >> fichier 2>&1

les nouvelles données (**stdout et stderr**) seront inséré à la fine du fichier

```
user1@PC:~$ cat fichier  
PC  
user1@PC:~$ { pwd ; HOSTNAME; } > fichier 2>&1  
user1@PC:~$ cat fichier  
PC  
/home/user1  
HOSTNAME: commande introuvable
```

Redirection le canal d'entrée vers une commande

Les caractères <, << ou (0<, 0<<) permettent rediriger l'entrée standard **stdin (fichier ou clavier)** vers une commande **commande < fichier**

L'entrée standard (**stdin dans ce cas un fichier**) sera redirigé vers une

```
user1@PC:~$ wc< fichier  
8 24 208
```

commande

commande << FIN

L'entrée standard (**stdin dans ce cas le clavier**) sera redirigé vers une

comman

de FIN est une chaîne qui désigne fin de la saisie

```
user1@PC:~$ sort -n << FIN  
> 120  
> 10  
> 1  
> 30  
> FIN  
1  
10  
30
```

120