

# Quelques notions sur Linux

## Prise en main

Pour explorer le monde de Linux, nous allons utiliser un émulateur du système Linux qui nous donnera l'accès à une invite de commande :

<https://chinginfo.fr/weblinux>

Lors du lancement, vous aurez un écran d'accueil de la forme suivante :

```
devtmpfs: mounted
Freeing unused kernel memory: 160K
This architecture does not have kernel memory protection.
Run /sbin/init as init process
Run /etc/init as init process
Run /bin/init as init process
process '/bin/busybox' started with executable stack
init started: BusyBox v1.28.4 (2018-06-01 07:15:55 UTC)
random: fast init done
ethoc 92000000.ethoc eth0: Link is Up - 10Mbps/Full - flow control off
udhcpd: started, v1.28.4
Setting IP address 0.0.0.0 on eth0
udhcpd: sending discover
udhcpd: sending select for 10.5.143.96
udhcpd: lease of 10.5.143.96 obtained, lease time 900
Setting IP address 10.5.143.96 on eth0
Deleting routers
route: SIOCDELRT: No such process
Adding router 10.5.0.1
Recreating /etc/resolv.conf
Adding DNS server 10.5.0.1
Adding DNS server 8.8.8.8
Adding DNS server 8.8.4.4
Console Linux / login:
```

Les différents utilisateurs et mot de passe pour la connexion sont inscrites dans la page. Pour nous identifier avec l'utilisateur *alice*, nous allons saisir :

- d'abord saisir le login *alice*
- valider le choix en appuyant sur "Enter"
- saisir le mot de passe *alice22* (*attention rien n'apparaît à l'écran lors de la saisie du mot de passe*)
- valider la saisie en appuyant sur "Enter"

```
Console Linux / login: alice
Password:
Authentification réussie.

                _   _   _
               / \ / \ / \
              |___|___|___|___|___|___|___|___|
              / \ / \ / \
             /___/___/___/___/___/___/___/___/
alice:~$
```

Nous voyons que l'authentification a réussi en observant la ligne *alice:~\$* :

- *alice* : indique le nom de l'utilisateur courant
- *~* : indique le répertoire courant. Ici "*~*" indique le répertoire personnel de l'utilisateur courant (*ici, alice*).
- *\$* : permet de séparer le préambule, des instructions saisies par l'utilisateur.

## L'utilisateur courant

La commande *id* permet de connaître l'utilisateur courant (*ici, alice*) et le dossier courant (*ici, /home/alice*) où seront exécutés les instructions de l'utilisateur :

```
alice:~$ id
uid=1002(alice) gid=2002(alice) groups=2002(alice),3000(elevés)
alice:~$ pwd
/home/alice
alice:~$
```

Plus précisément, la commande `id` nous donne des informations plus précises sur l'utilisateur courant :

- `uid=1002 (alice)` : l'utilisateur courant est "alice" et son numéro d'identification est 1002
- `groups=1002(alice), 3000(elevés)` : l'utilisateur courant appartient aux groupes `alice` et `elevés`

## Explorer les dossiers

La commande `cd` (*change directory*) permet, comme son nom l'indique, de changer de répertoire. La commande `ls` (*list*) permet de lister le contenu du répertoire courant :

```
alice:~$ cd ..
alice:/home$ ls
alice bob eve rep_a rep_b user
alice:/home$ cd rep_b
-bash: cd: rep_b: Permission denied
alice:/home$ cd rep_a
alice:/home/rep_a$
```

- l'instruction "`cd ..`" permet de remonter d'un dossier dans l'arborescence des fichiers. Voilà pourquoi le préfixe du terminal passe :  
de "alice:~\$" à "alice:/home"  
Le dossier courant était `~`, c'est-à-dire `/home/alice`. En remontant d'un niveau, on arrive dans le dossier `/home`
- La commande `ls` liste le contenu du dossier courant qui est maintenant `/home`. Nous voyons que le dossier `/home` contient lui-même cinq autres dossiers (*car affichés en bleu*):  
`alice ; bob ; eve ; rep_a ; rep_b ; user`

Nous remarquons :

- ➔ que l'utilisateur `alice` n'a pas les droits pour accéder au dossier `/home/rep_a`,
- ➔ alors qu'elle peut accéder au dossier `/home/rep_b`.

## L'utilisateur root

**Remarque :** dans tous les systèmes d'exploitation Linux, il existe un utilisateur `root`, appelé également "super-utilisateur", qui est l'administrateur du système d'exploitation.

Il a pour particularité de pouvoir accéder à tous les fichiers, de changer les droits d'accès, les mots de passe de tous les utilisateurs.

Son numéro d'identification est toujours 0.

On s'authentifie à l'aide de la commande `su` (*abréviation de "super user"* qui permet de changer d'utiliser l'utilisateur `root` dans le terminal :

```
alice:/home/rep_a$ su
Password:
Authentification réussie.
```



```
root:~#
```

Nous remarquons que l'invite de commande s'est modifiée. Particulièrement son préfixe :  
de `alice:/home/rep_a$` vers `root:~#`

Nous contrôlons l'identité et quelques caractéristiques de l'utilisateur à l'aide de la commande `id` et le dossier courant (*qui est le dossier personnel de l'utilisateur root*) à l'aide de la commande `pwd` :

```
root:~# id
uid=0(root) gid=0(root) groups=0(root)
root:~# pwd
/root
root:~#
```

On remarque :

- que l'identifiant de l'utilisateur `root` est 0 (`uid=0(root)`) et que son groupe est le groupe `root` (`groups=0(root)`)
- son dossier personnel est un dossier nommé `root` situé à la racine `/` de l'arborescence de fichiers.

## Gestion des utilisateurs

Avec Linux, la plupart des configurations du système passe par des fichiers textes que seul l'utilisateur `root` a le droit de consulter et de modifier.

Pour la gestion des utilisateurs, ce sera le fichier `/etc/passwd` et pour la gestion des groupes d'utilisateurs, ce sera le fichier `/etc/group`

Pour accéder au fichier "`passwd`", nous utilisons l'instruction "`cd /etc`" pour se rendre dans son dossier parent, puis nous utilisons la commande `ls` pour vérifier son existence :

```
root:~# cd /etc
root:/etc# ls
X11                group              joe                pointercal         ts.conf
dillo              host.conf          lynx               resolv.conf        udev
directfbrc        inetd.conf         network           services           xinitrc
fonts              init.d             nsswitch.conf     ssh
fstab              inittab            passwd            ssl
root:/etc#
```

Nous allons éditer le fichier `passwd` à l'aide de l'instruction `nano`.

```
root:/etc# nano passwd
```

Voici le contenu du fichier `/etc/passwd` affiché. En modifiant la fin de la ligne d'`alice` :  
de `/usr/bin/bash` vers `/bin/false`

Cette modification empêchera à l'utilisateur `alice` de s'authentifier.

```
GNU nano 2.2.6      File: passwd      Modified

root:WcOu3KdV.U4EM:0:0:0:/:root:/usr/bin/bash
sshd:*:27:27:sshd privsep:/var/empty:/sbin/nologin
user:wSVM0B9k1XrXA:1000:0:Linux User,,,:/home/user:/usr/bin/bash
bob:wItD6lTqmKUps:1001:2001:Linux User,,,:/home/bob:/usr/bin/bash
alice:n8x6Xq69z4vJg:1002:2002:Linux User,,,:/home/alice:/bin/false
eve:IQft.0tj0mvM6:1003:2003:Linux User,,,:/home/eve:/usr/bin/bash
nobody:*:65534:65534:Nobody:/tmp:/bin/false

^G Get Help      ^O WriteOut     ^R Read File    ^Y Prev Page    ^K Cut Text      ^C Cur Pos
^X Exit          ^J Justify      ^W Where Is     ^V Next Page    ^U UnCut Text   ^T To Spell
```

Nous quittons le logiciel `nano` par la combinaison de touches `Ctrl+X` et après enregistrement, nous devons voir :

```
root:/etc#
```

Avant de vérifier que l'effet de cette modification est immédiat pour l'utilisateur `alice`, nous allons afficher les groupes d'utilisateurs utilisés par le système d'exploitation.

Ces informations sont contenues dans le fichier `/etc/group`. Utilisons la commande `cat` pour afficher le contenu du fichier directement dans le terminal :

```
root:~# cd /etc
root:/etc# cat group
root:x:0:root
user:x:2000:
bob:x:2001:
alice:x:2002:
eve:x:2003:
eleves:x:3000:bob,alice,eve
professeurs:x:4000:eve
nogroup:x:65534:
root:/etc#
```

Nous remarquons :

- chaque utilisateur (`alice`, `bob`, `eve`, `user`, `root`) possède un groupe à leur nom.
- et qu'il existe trois groupes : `eleves`, `professeurs` possédant des utilisateurs alors `nogroup` ne possède pas d'utilisateur.

Maintenant, vérifions qu'`alice` ne peut plus s'authentifier dans le terminal. La commande `exit` permet de fermer la session courante.

Nous allons l'utiliser deux fois : une fois pour fermer la session de `root`, une seconde fois pour fermer celle d'`alice`.

Nous revenons ici à l'invite d'authentification :

```
root:/etc# exit
exit
alice:/home/rep_a$ exit
logout
Console Linux / login:
```

Malgré nos tentatives, il n'est pas possible de s'authentifier sous l'utilisateur `alice` :

```
Console Linux / login: alice
Password:
Console Linux / login:
```

**Remarque :** vous pouvez modifier tout ce que vous voulez sur cet émulateur. Il suffit de rafraîchir la page pour revenir à la configuration d'origine du système d'exploitation.

## • Droit d'accès aux fichiers •

Nous avons vu que l'utilisateur `alice` n'avez pas accès au dossier. Ainsi, le système d'exploitation refuse l'accès à certains dossiers et fichiers aux utilisateurs. Regardons ce qui se passe :

- Pour l'utilisateur `alice` :

```
alice:~$ cd /home
alice:/home$ cd rep_a
alice:/home/rep_a$ touch fichierC.txt
touch: fichierC.txt: Permission denied
alice:/home/rep_a$ cd ..
alice:/home$ cd rep_b
-bash: cd: rep_b: Permission denied
alice:/home$
```

Nous voyons que :

- ➔ `alice` accède au dossier `rep_a` mais ne peut pas créer un nouveau fichier (*nommé* `fichierC.txt`) à l'aide de la commande `touch`.
- ➔ l'accès au dossier `rep_b` lui est refusé.

- Pour l'utilisateur `eve` :

Commençons à se logger au terminal avec l'utilisateur `eve` :

```
Console Linux / login: eve
Password:
Authentification réussi.

      ()_()()_()
       ..)(..
        ( | | )
         *  *
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
<_ , - - , _ | / | _ >

eve:~$
```

Regardons maintenant les droits d'accès de eve à ces fichiers:

```
eve:~$ cd /home
eve:/home$ cd rep_a
eve:/home/rep_a$ touch fichierA.txt
touch: fichierA.txt: Permission denied
eve:/home/rep_a$ cd ..
eve:/home$ cd rep_b
eve:/home/rep_b$ touch fichierB.txt
eve:/home/rep_b$ ls -l
total 1
-r--r--r--  1 nobody   nogroup   20 Jan 23 05:48 README.txt
-rw-r--r--  1 eve      eve       0 Jan 23 06:08 fichierB.txt
eve:/home/rep_b$
```

Nous voyons que:

- ➔ l'utilisateur eve a accès au dossier rep\_a mais n'a pas le droit de créer un nouveau fichier.
- ➔ par contre, elle peut accéder au dossier rep\_b et y créer un nouveau fichier (*nommé fichierB.txt*) à l'aide de la commande touch.

## Un début d'explication :

- La commande `id` pour connaître les groupes auxquels appartiennent les utilisateurs `alice` et `eve` :

```
eve:/home$ id eve
uid=1003(eve) gid=2003(eve) groups=2003(eve),3000(eleves),4000(professeurs)
eve:/home$ id alice
uid=1002(alice) gid=2002(alice) groups=2002(alice),3000(eleves)
```

Nous pouvons déjà déduire :

- ➔ le groupe `eleves` contient les deux utilisateurs `alice` et `eve`.
- ➔ le groupe `professeurs` contient l'utilisateur `eve` mais pas l'utilisateur `alice`.
- la commande "`ls -l`" permet d'avoir des informations complémentaires sur les fichiers et sous-dossiers du dossier courant. Voici le contenu du dossier `/home` :

```
eve:/home$ ls -l
total 3
drwxr-x---  6 alice    alice    0 Jan 23 05:48 alice
drwxr-xr-x  9 bob      bob      0 Jan 23 05:48 bob
drwx----- 5 eve      eve      0 Jan 23 05:48 eve
drwxr-x---  3 nobody  eleves   85 Jan 23 05:48 rep_a
drwxrwx---  4 nobody  professe 121 Jan 23 05:48 rep_b
drwxr-x---  4 user    user     0 Jan 23 05:48 user
```

En regardant plus précisément les droits des dossiers `rep_a` et `rep_b` :

```
drwxr-x---  3 nobody  eleves   85 Jan 23 05:48 rep_a
drwxrwx---  4 nobody  professe 121 Jan 23 05:48 rep_b
```

- on voit que le dossier `rep_a` est associé au groupe "`eleves`" : les utilisateurs `alice` et `eve` font partie de ce groupe.
- on voit que le dossier `rep_b` est associé au groupe "`professeurs`" : ce qui permet de justifier que les droits de `alice` et `eve` sont différents.
- dans la définition des droits de ces deux répertoires :  
`rep_a` : `drwxr-x---` et `rep_b` : `drwxrwx---`  
L'absence d'un "`w`" (*write*) dans la définition des droits de `rep_a` justifiera l'interdiction au groupe `eleve` de créer des fichiers dans ce dossier.