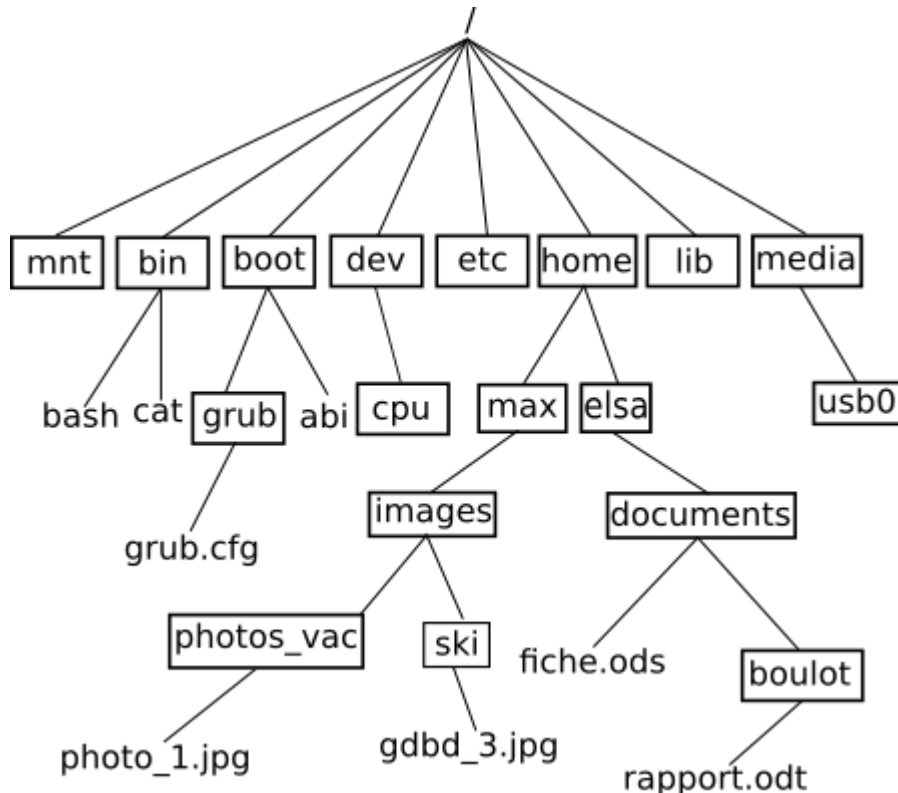


Chapitre E.3 - Les commandes LINUX

I. Le système de fichier LINUX

Dans la philosophie UNIX, dont découle le noyau Linux, toutes les ressources de l'ordinateur sont des fichiers. Une grande partie des commandes permettent de travailler sur les fichiers et les répertoires du système d'exploitation.

Le schéma ci-dessous représente une arborescence de fichiers Linux



Le répertoire / représente le premier répertoire de l'arborescence. Il est appelé "racine".

A. Le chemin absolu

Le chemin absolu d'un fichier ou d'un répertoire correspond au chemin à réaliser depuis la racine jusqu'à son emplacement. Un chemin absolu commence toujours par le caractère / qui correspond à la racine puis chaque répertoire traversé, séparé par le caractère /.

Par exemple, pour accéder au fichier "rapport.odt", on passe par les répertoires : home, elsa, documents et boulot. Le chemin absolu de ce fichier est donc :

```
/home/elsa/documents/boulot/rapport.odt
```

B. Le chemin relatif

Le chemin relatif d'un fichier ou d'un répertoire correspond au chemin à réaliser depuis un autre répertoire du système de fichiers. Un chemin relatif contient chaque répertoire traversé, séparé par le caractère /.

Par exemple, depuis le répertoire "elsa", pour accéder au fichier "rapport.odt" on passe par les répertoires : documents et boulots. Le chemin relatif du fichier "rapport.odt" depuis le répertoire "elsa" est :

Pour savoir si un chemin est un chemin absolu ou un chemin relatif on regarde le premier caractère :

- Si le premier caractère est un `/`, il s'agit d'un chemin absolu
- Si le premier caractère n'est pas un `/`, il s'agit d'un chemin relatif

C. Chemin spéciaux

Il est parfois nécessaire de faire référence au répertoire dans lequel on se situe. Le chemin actuel est représenté par le caractère `..`.

Ainsi depuis le répertoire "documents", le chemin

```
./fiche.ods
```

désigne le document "fiche.ods"

Il est parfois nécessaire de faire référence au répertoire parent du répertoire dans lequel on se situe. Le chemin du répertoire parent est représenté par les caractères `..`.

ainsi depuis le répertoire "elsa", le chemin

```
../max/images/photos_vac/photo_1.jpg
```

désigne le fichier "photo_1.jpg".

II. Les commandes de bases Linux

Commande	Description	Utilisation de base	Options courantes
<code>ls</code>	Liste les fichiers et répertoires	<code>ls</code>	<code>-l</code> (détails), <code>-a</code> (inclut les fichiers cachés)
<code>cd</code>	Change de répertoire	<code>cd</code> <code>/chemin/vers/dossier</code>	<code>cd ..</code> (remonte d'un niveau) <code>cd</code> (renvoie vers le répertoire personnel de l'utilisateur)
<code>pwd</code>	Affiche le répertoire courant	<code>pwd</code>	
<code>cp</code>	Copie des fichiers ou répertoires	<code>cp fichier1 fichier2</code>	<code>-i</code> (demander confirmation)
<code>mv</code>	Déplace ou renomme des fichiers ou répertoires	<code>mv fichier1 fichier2</code>	<code>-i</code> (demander confirmation)
<code>rm</code>	Supprime des fichiers ou répertoires	<code>rm fichier</code>	<code>-r</code> (récursif), <code>-f</code> (force), <code>-i</code> (demander confirmation)
<code>touch</code>	Crée un fichier vide ou met à jour la date de modification	<code>touch fichier</code>	
<code>mkdir</code>	Crée un répertoire	<code>mkdir dossier</code>	<code>-p</code> (crée les répertoires parents si nécessaire)

Commande	Description	Utilisation de base	Options courantes
<code>rmdir</code>	Supprime un répertoire vide	<code>rmdir dossier</code>	
<code>cat</code>	Affiche le contenu d'un fichier	<code>cat fichier</code>	<code>-n</code> (numérote les lignes), <code>-b</code> (numérote les lignes non vides)
<code>man</code>	Affiche le manuel d'une commande	<code>man commande</code>	<code>-k</code> (cherche dans la base de données des manuels)
<code>shutdown</code>	Éteint ou redémarre le système	<code>shutdown -h now</code>	<code>-r</code> (redémarre), <code>-h</code> (arrête le système)
<code>reboot</code>	Redémarre le système	<code>reboot</code>	-
<code>ping</code>	Vérifie la connectivité réseau	<code>ping adresse</code>	<code>-c</code> (nombre de requêtes), <code>-i</code> (intervalle)
<code>ifconfig</code>	Affiche ou configure les interfaces réseau	<code>ifconfig</code>	<code>-a</code> (affiche toutes les interfaces)
<code>zip</code>	Crée une archive ZIP	<code>zip archive.zip fichier</code>	
<code>unzip</code>	Décompresse une archive ZIP	<code>unzip archive.zip</code>	<code>-l</code> (affiche la liste des fichiers)
<code>grep</code>	Recherche un texte dans un fichier	<code>grep "mot" fichier</code>	<code>-r</code> (récursif), <code>-i</code> (insensible à la casse)
<code>sudo</code>	Permet d'exécuter une commande avec les privilèges administrateur	<code>sudo commande</code>	

III. Gestion des droits des utilisateurs

Le système de droits et de permissions sous UNIX est un aspect fondamental de la gestion de la sécurité du système.

A. Les utilisateurs

Du point de vue d'un fichier ou d'un répertoire, LINUX divise les utilisateurs en trois type :

- L'utilisateur (user) – Le propriétaire du fichier
- Le groupe (group) – Le groupe propriétaire du fichier
- Le reste du monde (others) – Les autres utilisateurs

B. Lecture de droits

En utilisant la commande `ls -l` on obtient un affichage comme celui-ci dessous :

```
-rwxr-xr-- roza staff 0 6 mai 11:56 monfic.sh
```

La partie `-rwxr-xr--` indique les droits des utilisateurs sur le fichier. Celle-ci se lit en omettant le premier `-`.

Le premier élément contient un `-` si l'élément est un fichier et contient un `d` si l'élément est un dossier

On divise l'élément en trois parties :

- `rwx` (contient les droits de l'utilisateur propriétaire du fichier)
- `r-x` (contient les droits du groupe propriétaire du fichier)
- `r--` (contient les droits de tous les autres utilisateurs)

Chaque partie est elle-même composée de trois éléments :

- `r` droit de lecture sur le fichier (read)
- `w` droit d'écriture sur le fichier (write)
- `x` droit d'exécuter le fichier (execute)

C. Modifier les droits des utilisateurs

La commande `chmod` permet de modifier les droits d'un fichier.

Pour utiliser la commande `chmod`, on peut spécifier les droits en utilisant deux méthodes principales :

1. Méthode symbolique

La méthode symbolique permet de modifier les droits en précisant les utilisateurs concernés et les droits à ajouter ou retirer. La syntaxe est la suivante :

```
chmod [utilisateur][opération][droits] fichier
```

- **Utilisateur :**
 - `u` : utilisateur propriétaire (user)
 - `g` : groupe propriétaire (group)
 - `o` : autres utilisateurs (others)
 - `a` : tous les utilisateurs (all)
- **Opération :**
 - `+` : ajoute un droit
 - `-` : retire un droit
 - `=` : remplace les droits existants
- **Droits :**
 - `r` : lecture (read)
 - `w` : écriture (write)
 - `x` : exécution (execute)

Exemples :

- Ajouter le droit d'exécution pour tous les utilisateurs :

```
chmod a+x script.sh
```

- Retirer le droit d'écriture pour le groupe :

```
chmod g-w fichier.txt
```

- Donner uniquement le droit de lecture à tous les utilisateurs :

```
chmod a=r fichier.txt
```

2. Méthode numérique (octale)

La méthode numérique utilise une valeur octale pour représenter les droits. Chaque droit est représenté par un chiffre :

- Lecture (**r**) : 4
- Écriture (**w**) : 2
- Exécution (**x**) : 1

On additionne les valeurs pour chaque catégorie d'utilisateur (propriétaire, groupe, autres) :

- **7** : lecture + écriture + exécution (4+2+1)
- **6** : lecture + écriture (4+2)
- **5** : lecture + exécution (4+1)
- **4** : lecture seule
- **0** : aucun droit

La syntaxe est la suivante :

```
chmod [valeur] fichier
```

Exemples :

- Donner tous les droits au propriétaire, lecture et exécution au groupe, et aucun droit aux autres :

```
chmod 750 fichier.txt
```

- Donner uniquement le droit de lecture à tous les utilisateurs :

```
chmod 444 fichier.txt
```

2. Méthode numérique (octale)

La méthode numérique utilise une valeur octale pour représenter les droits. Chaque droit est représenté par un chiffre :

- Lecture (**r**) : 4
- Écriture (**w**) : 2
- Exécution (**x**) : 1

On additionne les valeurs pour chaque catégorie d'utilisateur (propriétaire, groupe, autres) :

- **7** : lecture + écriture + exécution (4+2+1)
- **6** : lecture + écriture (4+2)
- **5** : lecture + exécution (4+1)
- **4** : lecture seule
- **0** : aucun droit

La syntaxe est la suivante :

```
chmod [valeur] fichier
```

Exemples :

- Donner tous les droits au propriétaire, lecture et exécution au groupe, et aucun droit aux autres :

```
chmod 750 fichier.txt
```

- Donner uniquement le droit de lecture à tous les utilisateurs :

```
chmod 444 fichier.txt
```

D. Vérifier les droits

Pour vérifier les droits d'un fichier ou d'un répertoire, utilisez la commande :

```
ls -l fichier
```

Cela affichera les droits actuels sous forme symbolique.

E. Changer le propriétaire ou le groupe

Pour modifier le propriétaire ou le groupe d'un fichier, utilisez les commandes suivantes :

- Changer le propriétaire :

```
chown utilisateur fichier
```

- Changer le groupe :

```
chgrp groupe fichier
```

- Changer à la fois le propriétaire et le groupe :

```
chown utilisateur:groupe fichier
```

Exemple :

- Attribuer le fichier **rapport.txt** à l'utilisateur **elsa** et au groupe **staff** :

```
chown elsa:staff rapport.txt
```

Remarque : Ces commandes nécessitent souvent des privilèges administratifs. Utilisez **sudo** si nécessaire.