

0.1 Gestion des droits

Linux est un système multi-utilisateur. Cela signifie que plusieurs personnes peuvent utiliser l'ordinateur simultanément (et pas uniquement les unes à la suite des autres), et que le système se charge de faire respecter la loi entre elles. Les ressources de la machine sont ainsi partagées équitablement, tant au niveau de la puissance de calcul qu'au niveau de la mémoire, du disque, des imprimantes... Cela implique une gestion avancée de la sécurité, qui est assurée par le noyau et par le système de fichiers.

- **Au niveau du noyau** : chaque utilisateur est identifié de manière unique par un numéro (uid) dans le système. Ce numéro est utilisé pour vérifier les droits de l'utilisateur, autrement dit, ce qu'il peut faire. Les droits des utilisateurs comprennent la possibilité de lire ou écrire un fichier, d'accéder ou non à une ressource ou d'exécuter un programme. Il est également possible de créer un ou plusieurs "groupes" d'utilisateurs, et de donner des droits particuliers à ces groupes. Tous les utilisateurs qui font partie de ce groupe recevront les droits du groupe.
- **Au niveau système de fichier** : chaque fichier linux est associé à un descripteur (i-node) qui contient un certain nombre de renseignements : identification du propriétaire, adresse des blocs utilisés par le fichier, sa taille en octets, son type, sa date de création ... La commande linux : **ls -l** permet d'en avoir un aperçu des droits

```
ls -l /mnt/multimedia/test.mp3
-rwxrwxrwx 1 kernel users 3.8M jun 2 2001 /mnt/multimedia/test.mp3
```

0.1.1 1. Le Type de fichier :

C'est le "-" qui indique qu'il s'agit d'un fichier ordinaire. Linux distingue 4 types de fichiers :

- **Les fichiers ordinaires (-)** : Se sont tous les fichiers courants, .txt, .doc, .mp3 .jpeg ou même sans extensions.
- **Les fichiers spéciaux (c ou b)** : Se sont les fichiers associés aux périphériques ils sont localisés dans /dev/
- **Les répertoires (d)** : Un répertoire est un fichier qui permet de référencer d'autres fichiers. Il est constitué d'une liste de couples (nom, index d'un i-node) qui permet d'accéder à un fichier à partir de son nom. Il possède en particulier 2 couples : "." lui-même et ".." son répertoire père. Petite exception pour / pour qui "." et ".." font référence à lui-même.
- **Les liens (l)** : Un lien permet d'accéder à un même fichier ou répertoire en utilisant un nom différent.

0.1.2 2. Les droits "classiques" :

Les droits "classiques" sont représentés par "rwxrwxrwx" de l'exemple plus haut. Vous remarquerez qu'il y'a 3 fois la même série de lettres : rwx. Les droits d'accès à un fichier vont être donnés :

- Au **propriétaire** du fichier, à qui correspondent les 3 premières lettres. Dans mon exemple plus haut l'utilisateur était "kernel".
- A un des **groupes** auquel appartient le propriétaire (les 3 lettres du milieu), Dans mon exemple plus haut le groupe était "users"
- A tous les **autres utilisateurs** (les 3 dernières lettres)

Ces droits se décomposent en 3 niveaux :

- **Accès en lecture (r)** : les utilisateurs autorisés peuvent visualiser le fichier ou le répertoire, ils peuvent copier le fichier vers un autre répertoire (la copie devient alors leur propriété).
- **Accès en écriture (w)** : les utilisateurs autorisés peuvent modifier le fichier ou le détruire. Dans le cas d'un répertoire ils peuvent manipuler (créer, copier, déplacer ou détruire) son contenu.
- **Accès en exécution(x)** : les utilisateurs autorisés peuvent exécuter le fichier. Si vous êtes un nouvel arrivant sur la banquise, cette notion est très importante. Il n'y a pas de .exe sous linux/unix, pour s'exécuter un programme a besoin d'avoir le bit exécutable à 1.

Tous ces droits sont évidemment géré par des bits codés dans l'i-node du fichier. Si le bit d'un des 9 champs est à 1 alors la lettre correspondante est affichée sinon c'est "-". Quelques exemples :

- **-rwxr-xr-x** : accès plein droit pour l'utilisateur, lecture/exécution pour les membres du groupe de l'utilisateur idem pour les autres.
- **-r-----** : accès en lecture seule pour l'utilisateur et accès interdit aux reste du monde (sauf root, c'est qui root ?)

Comme vous le savez il n'y a pas de règle sans exception, dans notre cas l'exception c'est root, mais c'est qui root ? root c'est l'administrateur mais en réalité c'est "**Dieu**". Il a à peu près tous les droits même celui du tuer le système. Vous devez donc être très gentil avec lui, sinon gare à sa colère :-). Si vous avez la possibilité de vous loguer en root, je vous conseille de limiter cela au tâches d'administration du système, une erreur est si vite arrivée.

0.1.3 3. La commande chmod :

La commande chmod permet de modifier les droits d'accès à un fichier. Un utilisateur ne peut modifier les droits que des fichiers qui lui appartiennent. Sauf root bien-sûr. Les droits sous linux sont repérés par une série de 3 valeurs octales. La première correspond au propriétaire, la seconde au groupe associé et la dernière au reste de la planète. Pour chaque catégorie d'utilisateur l'accès en lecture vaut à 4, l'accès en écriture vaut à 2 et l'accès en exécution 1. Quelques exemples :

- Pour avoir -rwxrwxrwx sur test.mp3 je dois taper :

```
chmod 777 test.mp3
```
- Pour avoir -rwxr-xr-x sur test.mp3 je dois taper :

```
chmod 755 test.mp3
```
- Pour avoir -r----- sur test.mp3 je dois taper :

```
chmod 400 test.mp3
```

0.1.4 4. Les droits spéciaux :

Trois fois trois font neuf, le compte est bon me direz-vous ? vous auriez tort, car en réalité c'est trois fois quatre qui font douze. Il existe sous linux des droits spéciaux. C'est ce qui permet par exemple que dans le répertoire /tmp auquel tout le monde a un accès plein droit, vous ne pouviez néanmoins supprimer les fichiers des autres utilisateurs.

0.1.5 4.1 Droits spéciaux sur les fichiers :

- **setuid** : Un fichier exécutable par son propriétaire, peut être setuid. C'est à dire qu'il s'exécute avec les droits de son propriétaire et non ceux de celui qui le lance. C'est d'ailleurs parce que "**passwd**" est setuid que vous pouvez modifier votre mot de passe avec "**passwd**" (ecrire dans /etc/passwd ou /etc/shadow).
- **setgid** : De la même façon, un exécutable peut être setgid, et s'exécuter avec les droits du groupe auquel il appartient.
- **sticky bit** : Enfin, un exécutable peut être "**sticky**" : cela signifie qu'il reste en mémoire même après la fin de son exécution, pour pouvoir être relancé plus rapidement. Attention seul root peut positionner le sticky bit.

0.1.6 4.2 Droits spéciaux sur les répertoires :

Il n'y a pas de setuid pour les répertoires, en revanche,

- **setgid** : Quand un répertoire est setgid, tous les fichiers créés dans ce répertoire appartiennent au même groupe que le répertoire. C'est utilisé par exemple quand plusieurs personnes travaillent sur un projet commun.
- **sticky bit** : Quand on positionne le sticky bit pour un répertoire, un utilisateur ne peut effacer que les fichiers qui lui appartiennent dans ce répertoire. C'est ce qui est utilisé pour le répertoire /tmp

0.1.7 4.3 chmod :

La logique est la même que précédemment, il suffit de rajouter une 4 ème valeur octale, telle que setuid vaille 4, setgid vaille 2 et sticky bit vaille 1. Si vous ne souhaitez pas jouer avec les droits spéciaux, faites un chmod avec 3 chiffres (par défaut les droits spéciaux conserveront leurs valeurs). Mais si vous souhaitez par exemple que le contenu du répertoire /mnt/projet appartienne au groupe propriétaire du répertoire mais qu'en sus chacun ne puisse que ses fichiers, vous pouvez taper :

```
chmod 3777 /mnt/projet
```

Notez que c'est la colonne la plus à gauche qui code les droits spéciaux.