Systèmes

Contient les cours et notions théoriques sur les systèmes d'exploitations.

- Linux - Gestion des permissions

Linux - Gestion des permissions

Difficulté : Novice

Notions : Unix, Linux, gestion de droits.

I. Introduction

En gestion de droits sur les systèmes Unix, il y a 3 notions importantes :

- quel est le type de l'objet
- qui est propriétaire de l'objet (owner/group)
- quelles sont les permissions de l'objet

Lorsque l'on gère les systèmes de fichier, si l'on fait une commande :

```
ls -al
```

```
root@52e1fe30fb74:/var/www/glpi# ls -al
total 344
drwxr-xr-x 22 www-data www-data  4096 Nov  5 10:06 .
drwxr-xr-x  1 root     root      4096 Nov  5 10:06 ..
-rw-r--r--  1 www-data www-data 89846 Nov  5 09:31 CHANGELOG.md
-rw-r--r--  1 www-data www-data  2060 Nov  5 09:31 CONTRIBUTING.md
-rw-r--r--  1 www-data www-data   682 Nov  5 09:31 INSTALL.md
-rw-r--r--  1 www-data www-data 35148 Nov  5 09:31 LICENSE
-rw-r--r--  1 www-data www-data  8174 Nov  5 09:31 README.md
-rw-r--r--  1 www-data www-data  1308 Nov  5 09:31 SECURITY.md
-rw-r--r--  1 www-data www-data   481 Nov  5 09:31 SUPPORT.md
drwxr-xr-x  4 www-data www-data  4096 Nov  5 09:31 ajax
-rw-r--r--  1 www-data www-data 60678 Nov  5 09:31 apirest.md
drwxr-xr-x  2 www-data www-data  4096 Nov  5 09:31 bin
drwxr-xr-x  2 www-data www-data  4096 Nov  5 10:06 config
```

Celui-ci est affiché de la façon suivante :

De gauche à droite :

- La première lettre représente de type de l'objet. Les lettres suivantes (rwx), les droits sur les fichiers.
  - `drwxr-xr-x`
- Le chiffre suivant représente le nombre de lien sur l'objet.
  - `2`
- Puis l'utilisateur propriétaire et le groupe propriétaire.
  - `www-data www-data`
- La taille du fichier (en octets).
- `35148`
- La date de dernière modification.
  - `Nov  5 09:31`
- Le nom de l'objet
  - `CHANGELOG.md`

>

> **Note** : dans cet exemple, l'utilisateur et le groupe ont le même nom mais sont bien deux éléments séparés.

II. Types d'objets

Sur linux, le principe de base est que tout est fichier.

C'est ensuite l'attribut de l'objet qui déterminera son type; Les plus utilisés sont :

- **-** : Par défaut, un fichier
- **d** : Un répertoire (directory)
- **s** : fichier de socket
- **l** : lien symbolique

Mais il existe aussi :

- **c** : Périphérique de type caractère
- **b** : Périphérique de type bloc
- **p** : Pipeline

Un socket est un type de fichier représentant un socket au niveau réseau.

Un lien symbolique est un lien logique permettant de cibler un répertoire / fichier qui se situe à un autre endroit afin de le rendre disponible dans le répertoire courant sans pour autant en créer un duplicata.

III. Propriété de l'objet

Les objets ont un couple de propriétaires.

Le premier est l'utilisateur propriétaire, le second est le groupe propriétaire.

Pour changer les propriétaires d'un objet, il faut utiliser la commande

```
chown <nomDuOwner>:<NomDuGroupe>
```

Il est possible d'ajouter ***-R*** pour rendre la modification récursive.

Exemple :

```
chown www-data:root index.php
```

> **Info** : Par défaut, lors de la création d'un objet, l'utilisateur ayant créé l'objet est défini en tant qu'utilisateur propriétaire et son groupe personnel en tant que groupe propriétaire.

IV. Droits de l'objet

4.1 Types de droits :

Les droits UNIX s'articulent autour de trois "actions" possibles :

- la lecture (r) : on peut par exemple lire le fichier avec un logiciel. Lorsque ce droit est alloué à un répertoire, il permet d'afficher la liste des fichiers du répertoire (la liste des fichiers présents à la racine de ce répertoire).
- l'écriture (w) : on peut modifier le fichier et le vider de son contenu. Lorsque ce droit est alloué à un répertoire, il autorise la création, la suppression et le changement de nom des fichiers qu'il contient, quels que soient les droits d'accès des fichiers de ce répertoire (même s'ils ne possèdent pas eux-mêmes le droit en écriture). Néanmoins le droit spécial *sticky bit* permet de passer outre ce comportement.

- l'exécution (x) : on peut exécuter le fichier s'il est prévu pour, c'est-à-dire si c'est un fichier exécutable. Lorsque ce droit est attribué à un répertoire, il autorise l'accès (ou ouverture) au répertoire.

4.2 Calculs des droits :

Ces lettres, appelés "flags" définiront les droits sur le répertoire. Ces droits sont basés sur un système en binaire sur 3 bits.

Lettre	**r**	**w**	**x**
Valeur binaire	*4*	*2*	*1*

Pour calculer tous les droits, il faudra additionner les valeurs de chaque bits pour en définir la valeur totale.

par exemple, lecture + écriture = r+w = 2+4 = 6

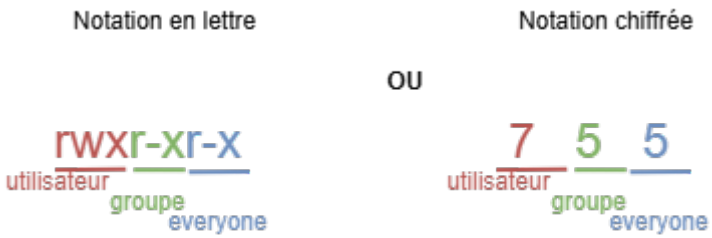
Ainsi, on aura la correspondance suivante :

Valeur Décimale	Droits
1	exécution (**--x**)
2	écriture (**--w**)
3	écriture + exécution (**-wx**)
4	lecture (**r--**)
5	lecture + exécution (**r-x**)
6	lecture + écriture (**rw-**)
7	tous les droits (**rwx**)

> **Attention** : la valeur 7 confère tous les droits et doit être appliquée en connaissance de cause, sous peine de représenter une faille de sécurité.

4.3 Représentation des droits :

Les droits sur les objets seront ensuite représentés sous ce format :



Le premier groupe de 3 lettres sont les permissions de l'utilisateur propriétaire.

Le second groupe représente les droits du groupe propriétaire.

Le 3ème groupe représente les droits appliqués à tout le monde.

4.4 Modification des droits :

Les permissions se modifient avec les commande ci-dessous :

```
chmod <notation décimale ou lettres> <objet>
```

Il est possible d'ajouter ***-R*** pour rendre la modification récursive.

Exemple :

```
chmod 755 /var/www/index.php
```

D'autres commandes existent pour modifier ou ajuster les droits, notamment avec les lettres ou pour ajouter/retirer un droit spécifique à l'utilisateur, au groupe ou à everyone.

Mais la méthode évoquée ci-dessus reste la plus simple et la plus rapide.

V. Droits spéciaux

5.1 Droits SUID :

Ce droit s'applique aux fichiers exécutables, il permet d'allouer temporairement à un utilisateur les droits du propriétaire du fichier, durant son exécution.

En effet, lorsqu'un programme est exécuté par un utilisateur, les tâches qu'il accomplira seront restreintes par ses propres droits, qui s'appliquent donc au programme.

Lorsque le droit SUID est appliqué à un exécutable et qu'un utilisateur quelconque l'exécute, le programme détiendra alors les droits du propriétaire du fichier durant son exécution.

Bien sûr, un utilisateur ne peut jouir du droit SUID que s'il détient par ailleurs les droits d'exécution du programme.

Ce droit est utilisé lorsqu'une tâche, bien que légitime pour un utilisateur classique, nécessite des droits supplémentaires (généralement ceux de *root*). Il est donc à utiliser avec précaution.

Pour des partitions supplémentaires, il faut activer le bit suid pour pouvoir l'utiliser en le spécifiant dans les options des partitions concernés dans le fichier fstab.

5.2 Droits SGID :

Ce droit fonctionne comme le droit SUID, mais appliqué aux groupes. Il donne à un utilisateur les droits du groupe auquel appartient le propriétaire de l'exécutable et non plus les droits du

propriétaire.

VI. Conclusion

Le système de permission Unix est assez simple et équivoque. Mais en contrepartie, il est assez limité.

Il n'est possible, en effet, d'attribuer nativement des autorisation que pour 'utilisateur, groupe, tous les autres'.

Si l'on veut avoir une gestion plus fine et plus complexe, par exemple pour des serveurs de fichiers, il faudra utiliser des logiciels complémentaire qui prendront en charge les listes de contrôle d'accès.