**Les permissions sur les fichiers et les répertoires**

Nous avons vu que le paramètre -l de la commande ls permet d'afficher au long les attributs des fichiers, dont les droits d'accès. Voyons en détails comment interpréter les droits d'accès.

Au début de chaque ligne générée par **ls -l**, nous retrouvons une suite de dix caractères.

Le premier caractère identifie le type du fichier.

Les types de fichier les plus courants sont les "fichiers" et les "dossiers".

* **"-" représente des fichiers ordinaires**

Voici des exemples de fichiers ordinaires

* fichier texte
* fichier binaire
* fichier image
* fichier compressé
* …

Commande pour créer un fichier ordinaire

**touch ou vi**

Comment afficher une liste de fichiers ordinaires

**ls -l | grep ^-**

* **"d" représente des dossiers**

Un dossier peut contenir des fichiers ordinaires, des dossiers et des fichiers spéciaux. La couleur des dossiers est bleu.

Commande pour créer un dossier

**mkdir**

Comment afficher une liste de dossiers

**ls -l | grep ^d**

**Tout fichier du système appartient à la fois à un utilisateur (son "propriétaire") et à un groupe.**

**EXEMPLE**

**-rw-rw-r-- 1 taylor root 280232 Oct 10 16:22 info.txt**

info.txt est donc un fichier ordinaire (-) appartenant à taylor, au groupe root.

Taylor, le propriétaire, a accès en lecture, en écriture, mais pas en exécution (rw-).

Le groupe root a les mêmes droits que Taylor (rw-).

Les autres usagers, n'ont accès qu'en lecture seulement (r--).

Nous verrons que LINUX interprète les droits sur les répertoires de façon un peu différente que pour les fichiers.

**Les autorisations pour les fichiers**

**R** le droit de lecture permet de lire le contenu du fichier

**W** le droit d'écriture permet de modifier le contenu du fichier

**X** le droit d'exécution est attribué à un script ou un programme exécutable

**Les autorisations pour les répertoires**

Pour les répertoires, la signification des attributs est différente de celle d'un fichier.

Les droits d'un répertoire ne concernent que le contenu directement sous ce répertoire.

**Pour comprendre les permissions sur un répertoire, il faut voir celui-ci comme un catalogue contenant le nom des fichiers et les inodes correspondants.**

**R** permet d'afficher la liste des fichiers / répertoires d'un répertoire

**Si on attribue R, il faut attribuer aussi X sur le répertoire.**

**W** permet d'ajouter, de renommer et de supprimer des fichiers / répertoires à l’intérieur d'un répertoire

**Si on attribue W, il faut attribuer aussi X sur le répertoire.**

**X** permet de se déplacer dans un répertoire pour accéder aux fichiers / répertoires du répertoire

**La valeur en octale des autorisations**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **utilisateur** | | | **groupe** | | | **others** | | |
| **r** | **w** | **x** | **r** | **w** | **x** | **r** | **w** | **x** |
| 400 | 200 | 100 | 40 | 20 | 10 | 4 | 2 | 1 |

**Lorsqu’un utilisateur tente d’accéder à un fichier, les permissions sont vérifiées de gauche à droite**

**Comment changer les permissions avec les valeurs octales**

chmod *valeur\_octale* chemin\_du\_fichier

chmod 777 /tmp/f1.txt attribue rwx à user, group et other

chmod 666 /tmp/f2.txt attribue rw à user, group et other

chmod 770 /tmp/f3.txt attribue rwx à user et group et rien à other

chmod 070 /tmp/f4.txt attribue rwx à group et rien à user et other

**Comment changer les permissions**

**chmod (notation symbolique)**

chmod [catégorie d'usager] [+-=] [droits] chemin\_du\_fichier

* les catégories: u (user) g (group) o (other) a (all)
* + (ajoute des droits) - (retire des droits) = (modifie les droits existants)
* droits: r w x

ajoute le droit d'exécution au propriétaire sur toto.txt

chmod u+x toto.txt

enlève le droit d'écriture au groupe sur toto.txt

chmod g-w toto.txt

modife les droits de tout le monde pour rw sur toto.txt

chmod a=rw toto.txt

modifie les droits du propriétaire et de son groupe à r et x et

modifie les droits de "other" afin qu'ils n'aient plus d'accès

chmod ug=rx,o= toto.txt

**La commande chown**

**chown (pour changer le propriétaire d'un fichier et changer le groupe)**

chown nom\_du\_propriétaire chemin\_du\_fichier

chown nom\_du\_propriétaire:nom du groupe chemin du fichier

**La commande chgrp**

**chgrp (pour changer le groupe)**

chgrp nom\_du\_groupe chemin\_du\_fichier

**Les autorisations étendues pour un fichier**

**SUID (fichier exécutable seulement)**

* lors de l'exécution, le processus aura les privilèges du propriétaire du programme et non pas ceux de l'utilisateur
* chmod u+s nom\_du\_fichier

**exemples:**

/bin/ping

/bin/mount

**Avant d'attribuer l'autorisation SUID à un fichier exécutable vous devez réfléchir à l'impact.**

**SGID (fichier exécutable seulement)**

L'attribut SGID (Set Group ID) fonctionne de manière similaire à l'attribut SUID, mais il agit sur le groupe plutôt que sur l'utilisateur. Lorsqu'un fichier est marqué avec l'attribut SGID, lorsqu'un utilisateur exécute ce fichier, le programme s'exécute avec les permissions du groupe propriétaire du fichier, plutôt qu'avec les permissions du groupe de l'utilisateur qui l'exécute.

* lors de l'exécution, le processus aura les privilèges du groupe du programme et non pas ceux de l'utilisateur
* chmod g+s nom\_du\_fichier

**Avant d'attribuer l'autorisation SGID à un fichier exécutable vous devez réfléchir à l'impact.**

**Les autorisations étendue pour un répertoire**

**SGID**

* un fichier créé à l’intérieur du répertoire aura le même groupe que le répertoire et non pas le groupe principal de son créateur

**exemple:**

/var/ftp/pub/

**sticky**

* empêche les utilisateurs d'effacer ou de renommer des fichiers s'ils ne sont pas les propriétaires de ces fichiers mais permet aux utilisateurs de modifier le contenu des fichiers

**La valeur en octale des autorisations étendues**

|  |  |  |
| --- | --- | --- |
| **SUID** | **SGID** | **sticky** |
| **s** | **s** | **t** |
| 4000 | 2000 | 1000 |

**Comment attribuer les permissions étendues**

**SUID**

* chmod u+s chemin\_du fichier

chmod u+s /tmp/toto.sh

**SGID**

* chmod g+s chemin\_du\_fichier

chmod g+s /tmp/tata.sh

* chmod g+s chemin\_du\_répertoire

chmod g+s /tmp/info/

**"sticky"**

* chmod o+t chemin\_du\_répertoire

chmod o+t /tmp/test/

**Représentation des permissions étendues**

Si on attribue **SUID** alors le x de l’utilisateur est remplacé par un s.

exemple: rw**s**r-xr-x

Si on attribue **SGID** alors le x du groupe est remplacé par un s.

exemple: rwxr-**s**r-x

Si on attribue "**sticky**" alors le x des autres est remplacé par un t.

exemple: rwxrwxrw**t**

Si ces autorisations sont attribuées et que l'autorisation d’exécution correspondant ne l’est pas alors les autorisations s'affichent en majuscules.

* rw**S**r-xr-x
* rwxr-**S**r-x
* rw-r- -r-**T**

1. **rwSr-xr-x** :
   * **rw-** : l'utilisateur (propriétaire) a la permission de lecture et d'écriture.
   * **Sr-** : l'attribut SUID est activé. Le fichier est exécutable avec les droits du propriétaire.
   * **xr-x** : les membres du groupe ont la permission d'exécuter le fichier, mais pas de le modifier. Les autres utilisateurs ont également la permission d'exécuter le fichier, mais pas de le modifier.
2. **rwxr-Sr-x** :
   * **rwx** : l'utilisateur a la permission de lecture, écriture et exécution.
   * **r-S** : l'attribut SGID est activé. Le fichier est exécutable avec les droits du groupe propriétaire.
   * **r-x** : les autres utilisateurs ont la permission de lire et d'exécuter le fichier, mais pas de le modifier.
3. **rw-r--r-T** :
   * **rw-** : l'utilisateur a la permission de lecture et d'écriture.
   * **r--** : les membres du groupe ont la permission de lecture uniquement.
   * **r-T** : l'attribut Sticky Bit est activé sur le répertoire. Cela signifie que seuls les propriétaires des fichiers dans le répertoire peuvent les supprimer, même si d'autres utilisateurs ont des droits d'écriture sur ce répertoire.

**Comment changer les permissions étendues avec les valeurs octales**

chmod *valeur\_octale* chemin\_du\_fichier

chmod 4777 /tmp/f1.sh attribue SUID sur un fichier qui a 777

chmod 2777 /tmp/f1.sh attribue SGID sur un fichier qui a 777

chmod 2777 /tmp/info/ attribue SGID sur un répertoire qui a 777

chmod 1777 /tmp/test/ attribue "sticky" sur un répertoire qui a 777