

# Gestion des droits d'accès aux fichiers

Emilien Macchi

Ubuntu est une distribution GNU/Linux (qui lui même fait partie de la famille des systèmes Unix). Ce système d'exploitation est donc capable de gérer de manière très puissante les droits d'accès aux fichiers. Sous Unix (donc sous Linux), tout est fichier, même un répertoire !

Les droits d'accès déterminent donc la possession d'un fichier ou d'un répertoire à un utilisateur, ou à un groupe d'utilisateurs. Ils précisent aussi les actions que ces utilisateurs ont le droit de faire sur un fichier selon qu'ils en sont propriétaires, ou sont membres du groupe propriétaire, ou ni l'un ni l'autre. Nous verrons plus tard ces actions.

Pour mieux comprendre comment gérer les droits sur des fichiers, nous verrons tout d'abord la notion de propriétaire, puis nous étudierons les différentes permissions qui s'appliquent sur un ou plusieurs fichiers et, enfin, nous verrons comment gérer ces droits d'accès graphique et en ligne de commande.

## La notion de « Propriétaire »

Quand un fichier ou un répertoire est créé, il appartient forcément à un utilisateur (réel ou système). Cet utilisateur sera le « propriétaire » du fichier.

La possession d'un fichier est caractérisée sur 3 niveaux :

- Le 1<sup>er</sup> niveau : *Propriétaire* du fichier ; souvent, il s'agit du créateur du fichier. Caractérisé par la lettre « u » (user, comme utilisateur).
- Le 2<sup>nd</sup> niveau : *Groupe Propriétaire* du fichier ; il est possible pour un utilisateur membre du groupe propriétaire du fichier d'avoir des permissions exclusives. Caractérisé par la lettre « g » (group, comme groupe).
- Le 3<sup>ème</sup> niveau : les *Autres* ; tout utilisateur qui n'est ni propriétaire, ni membre du groupe propriétaire fera partie des « Autres ». Caractérisé par la lettre « o » (other, comme autre).

Pour désigner les 3 en même temps, vous pouvez utiliser la lettre « a » (all, comme tous).

✓ **Note :** Tout fichier créé avec la commande `sudo` appartiendra au super-utilisateur « root ». Vous serez donc un jour où l'autre amené à changer le propriétaire du fichier pour pouvoir l'utiliser avec votre compte utilisateur).

La commande `chown` (change owner) qui est accessible par l'utilisateur propriétaire du fichier ou « root », permet de changer le propriétaire d'un fichier. Pour mieux comprendre, nous vous proposons 2 exemples :

- Le propriétaire du fichier « file » est l'utilisateur « pierre » :  
`sudo chown pierre file`
- Le fichier « file » appartient à l'utilisateur « pierre » et au groupe « equipe1 » :  
`sudo chown pierre:equipe1 file`

Pour le 1<sup>er</sup> exemple, nous changeons le propriétaire du fichier, alors que pour le 2<sup>nd</sup>, nous modifions aussi le groupe auquel appartient le fichier.



Figure 1. Permissions du fichier « file »

La commande `chgrp` (change group), accessible au propriétaire du fichier et à root, change le groupe auquel appartient le fichier.

La syntaxe est simple :

```
sudo chgrp equipe2 file
```

Le fichier « file » appartient maintenant au groupe « equipe2 ». Tous les utilisateurs de ce groupe auront accès au fichier selon les permissions du groupe (2<sup>ème</sup> niveau, rappelez-vous !).

Nous avons vu la notion de propriétaire, étudions maintenant les permissions sur les fichiers.

## Les permissions

Elles désignent ce que chaque utilisateur a le droit de faire sur un fichier. Rappelons-nous qu'il y a 3 niveaux d'utilisateurs, le propriétaire, le groupe propriétaire et les autres.

Les permissions sur les fichiers se caractérisent sur 3 niveaux :

- **Lecture** : nécessaire pour accéder au contenu d'un fichier. Pour un répertoire, elle permettra de lister son contenu. Elle est notée « r » (pour read, lire) et « 4 » en octal.

- Quelques exemples d'utilisation : écouter un fichier audio, lire un film, visualiser une documentation PDF, lire le contenu d'un fichier texte, etc...

- **Écriture** : nécessaire pour modifier ou modifier un fichier. Elle est notée « w » (pour write, écrire) et « 2 » en octal.

- Quelques exemples d'utilisation : Modifier un fichier texte, supprimer n'importe quel fichier dans un dossier, etc...

- **Exécution** : nécessaire précisément pour les logiciels, pour qu'ils puissent être exécutés. Pour un répertoire, elle accordera son ouverture. Elle est notée « x » (pour execute, exécuter) et « 1 » en octal.

- Quelques exemples d'utilisation : exécuter un logiciel, ouvrir un répertoire, traverser un répertoire dont nous ne pouvons pas visualiser le contenu.

Pour être sur que vous avez compris :

- Pour un fichier :

- *rw* : tous les droits sur le fichier,
- *rx* : lire, et exécuter le fichier,
- *rw* : lire, modifier, et supprimer le fichier.

- Pour un répertoire :

- *rw* : tous les droits sur le répertoire,
- *r* : liste seulement le contenu du dossier,
- *x* : ouvre le répertoire seulement. Utile pour traverser un répertoire dont nous n'avons pas le droit de lire le contenu,
- *rx* : ouvre et liste le contenu du répertoire.

Maintenant que nous avons compris les différentes permissions sur les fichiers et les répertoires, voyons comment les mettre en œuvre.

## Graphiquement

Vous avez installé Ubuntu, vous êtes dans l'environnement graphique GNOME, votre gestionnaire de fichiers et de dossiers est *Nautilus*.

Avec ce logiciel, vous avez la possibilité de visualiser et de gérer graphiquement les droits d'un clic droit sur le fichier ou le dossier, puis sur *Propriétés*. Déplacez-vous dans l'onglet *Permissions*.

Vous pouvez sélectionner dans le menu déroulant les droits en lecture et écriture ou lecture seule pour le *Propriétaire* et le *Groupe*. Pour les *Autres*, vous pouvez choisir entre lecture et écriture, lecture seule et aucun. Vous pouvez choisir le groupe auquel appartient le fichier (en général, le groupe du propriétaire).

Lorsqu'il s'agit d'un répertoire, pour le *Propriétaire* et *Groupe* vous pouvez choisir entre *Création et suppression des fichiers*, *Accès aux fichiers*, et *Lister seulement les fichiers*. Pour les *Autres*, vous pouvez choisir entre *Création et suppression des fichiers*, *Accès aux fichiers*, *Lister seulement les fichiers* et *Aucun*.

- ✓ **Notez** qu'il est possible d'appliquer les droits aux fichiers contenus dans ce répertoire.



**Figure 2.** Permissions sur le répertoire « public\_folder »

Nous retrouvons bien ici les différents droits étudiés auparavant.

### En ligne de commande

Reprenons ce que nous avons dit plus haut :

- lecture/écriture/exécution (read - 4 / write - 2 / execute - 1), nous obtenons : *rw*x ou 4 / 2 / 1
- propriétaire/groupe/autres (user/group/other), nous obtenons : *u*go

Rappelons la commande bien connue pour lister le contenu d'un répertoire :

```
ls -l
```

Étudions la première colonne composée de 10 caractères :

- nature du fichier :
  - « - » : fichier
  - « l » : lien
  - « d » : dossier
- les 9 suivants correspondent aux 3 droits de respectivement *user*, *group* et *other*

Explication par l'exemple : dans la Figure 3, nous avons le fichier « file » qui est de nature « fichier », modifiable et supprimable par le propriétaire et seulement lisible pour les utilisateurs membres du groupe propriétaire et les autres.

L'outil « *chmod* » (change mode) qui n'est accessible qu'au propriétaire du fichier ou root, est l'outil qui

permet de modifier les permissions sur un fichier ou un dossier.

Il peut s'employer de 2 manières.

#### Avec les chiffres

Pour cela, vous devez retenir le chiffre en octal qui correspond au droit (ça va aller, il n'y en a que 4 !) :

- r = 4
- w = 2
- x = 1
- - = 0

Pour chaque groupement de droits (vous souvenez vous d'« *ugo* » ?), nous aurons 1 chiffre.

Par exemple :

- Pour *rw*x, nous aurons : 4+2+1 = 7
- Pour *r-x* on aura : 4+0+1 = 5
- Pour *r--*, on aura : 4+0+0 = 4

Pour attribuer tous les droits au propriétaire puis les droits de lecture aux membres du groupe auquel appartient le fichier et aux autres, nous aurons 744.

Compris ? C'est parti !

```
chmod 744 file
```

Les droits que nous avons vus au début sont appliqués sur le fichier « file ». L'option « *-R* » attribuera récursivement aux fichiers contenu dans un répertoire, les droits voulus.

```
chmod -R 755 dossier_public
```

Tous les fichiers contenus dans « dossier\_public » auront les droits du dossier.

#### Avec les lettres

Pour les utilisateurs de cette méthode, vous devez retenir les lettres suivantes :

- lecture/écriture/exécution, nous obtenons : *rw*x ou 4 / 2 / 1
- propriétaire/groupe/autres, nous obtenons : *u*go

Nous avons 3 sortes de modifications :

- ajout : « + »
- suppression : « - »
- ne rien changer : « = »

Si vous avez retenu cela, nous pouvons continuer avec la pratique.

```
chmod u+w, g+x, o-x file
```

Nous avons ajouté le droit d'écriture au propriétaire, ajouter le droit d'exécution au groupe propriétaire et enlever le droit d'exécution aux autres.

```
chmod +x file.sh
```

Cette commande permettra à tout le monde d'exécuter le fichier *file.sh*.

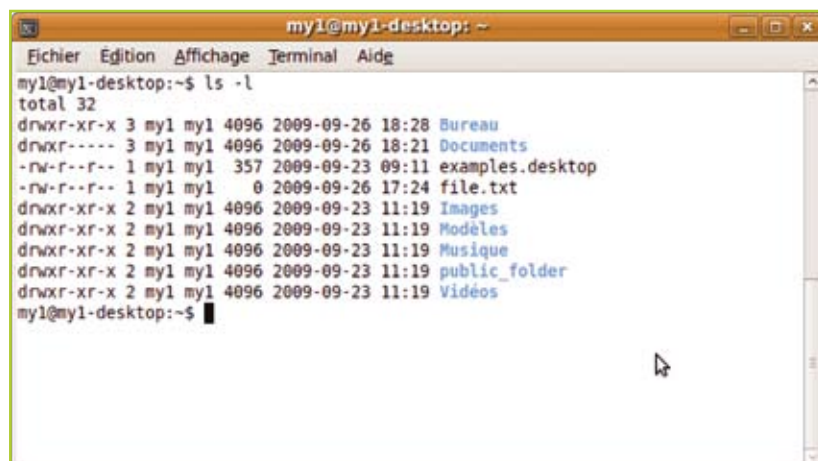
Graphiquement, ou en ligne de commande, nous avons vu qu'il est simple de gérer les droits sur les fichiers ou les répertoires.

Si vous êtes plusieurs utilisateurs sur une machine et que vous désirez contrôler les accès aux fichiers, il peut être intéressant d'appliquer ce genre de sécurité sur votre système de fichiers.



**Attention !** Ne modifiez jamais les droits sur les fichiers ou répertoires systèmes. Exemple : un « *chmod -R 755 /* » rendrait votre système d'exploitation complètement inutilisable.

Pour plus de détails, nous vous suggérons cet article sur Internet, qui détaille les Permissions sous Unix : [http://fr.wikipedia.org/wiki/Permissions\\_Unix](http://fr.wikipedia.org/wiki/Permissions_Unix).



```
myl@myl-desktop:~$ ls -l
total 32
drwxr-xr-x 3 myl myl 4096 2009-09-26 18:28 Bureau
drwxr----- 3 myl myl 4096 2009-09-26 18:21 Documents
-rw-r--r-- 1 myl myl 357 2009-09-23 09:11 exemples.desktop
-rw-r--r-- 1 myl myl 0 2009-09-26 17:24 file.txt
drwxr-xr-x 2 myl myl 4096 2009-09-23 11:19 Images
drwxr-xr-x 2 myl myl 4096 2009-09-23 11:19 Modèles
drwxr-xr-x 2 myl myl 4096 2009-09-23 11:19 Musique
drwxr-xr-x 2 myl myl 4096 2009-09-23 11:19 public_folder
drwxr-xr-x 2 myl myl 4096 2009-09-23 11:19 Vidéos
myl@myl-desktop:~$
```

Figure 3. Résultat de la commande