

# Module 11 – Droits sur les fichiers et répertoires

## Objectifs

- Comprendre le fonctionnement des droits
- Gérer les droits sur les répertoires et fichiers
- Gérer les utilisateurs et groupes propriétaires des répertoires et fichiers
- Gérer le modèle de droits
- Les droits spéciaux

Donner les droits suffisants, ni plus, ni moins

## Les droits Unix/Linux

### Les types de droits

- Les droits d'accès sur les fichiers sont directement inscrits dans l'inode des fichiers (répertoires). Ils sont découpés en trois colonnes : utilisateur propriétaire (user), groupe propriétaire (group) et les autres (other).
- En pratique, il y a toujours trois groupes de trois droits.

**rwxrwxr-x**  
utilisateur groupe other

Droit	Octal	Fichier	Répertoire
Lecture (r)	4	Permet de lire, ouvrir le fichier Ex : <code>cat</code>	Permet de lister le contenu du dossier. Ex : <code>ls</code>
Ecriture (w)	2	Permet d'écrire dans le fichier (le droit de lecture doit être présent)	Permet de modifier le contenu du dossier (créer, copier, renommer, supprimer) Ex : <code>mkdir cp rm</code>
Execution (x)	1	Permet d'exécuter le fichier binaire ou script shell (le droit de lecture doit être présent)	Permet à un utilisateur de traverser ou d'accéder à un fichier dans le dossier. Ex : <code>cd</code>

Pour calculer les droits en octal, il faut additionner les valeurs des droits présents pour les trois groupes

<code>rwX</code>	<code>=</code>	<code>4+2+1</code>	<code>= 7</code>
<code>rw-</code>	<code>=</code>	<code>4+2</code>	<code>= 6</code>
<code>r-X</code>	<code>=</code>	<code>4+1</code>	<code>= 5</code>
<code>r--</code>	<code>=</code>	<code>4</code>	

Exemples :

```

rwXrwXr-X = 775
rwXr-X--- = 750
rw-rw-r-- = 664

```

- Pour afficher les droits d'un fichier ou répertoire, la commande `ls` sera utilisée.

```

root@deb:~$ ls -l
total 16
-rw----- 1 user30 user30 616 09:34 Edition
drwxr-xr-x 2 user30 user30 4096 17:36 Nantes
lrwxrwxrwx 1 user30 user30 15 17:38 Quimper -> Rennes/Quimper/
drwxr-xr-x 3 user30 user30 4096 17:36 Rennes
-rw-r--r-- 1 user30 user30 33 17:35 todo.list

```

- Dans cet exemple, sont mis en avant les droits de l'utilisateur :

**propriétaire en vert**, les droits du **groupe propriétaire en bleu** et ceux des **autres en violet**.

## Modifier les droits et propriétaires

```
chmod <-R> [ugoa][+ - =][rwxX] fichier ou dossier
```

- La commande `chmod` permet de modifier les droits.
- Il est possible d'utiliser cette commande de deux façons :
  - Absolue : en octal ou symbolique avec le signe « `=` »
  - Relative : en symbolique avec les signes « `+` » ou « `-` »
- Attention à l'option `-R`, le fait de réaliser un `chmod` de façon récursive doit être effectué de façon réfléchie et ne doit pas être généralisé à tous les usages.

### Modification des droits

- Modifier les permissions d'un répertoire en notation symbolique

```
root@deb:~# chmod g+w,o-rx /data/commun
```

- Modifier les permissions d'un répertoire en notation octale

```
root@deb:~# chmod 770 /data/commun
```

- Modifier les permissions d'un répertoire ET de son contenu

```
root@deb:~# chmod -R g+w,o-rx /data/commun
```

## Modification du propriétaire et/ou du groupe propriétaire

```
chown <-R> [user]:[group] fichier(dossier)
```

- La commande **chown** permet de modifier l'utilisateur propriétaire et le groupe propriétaire
- Modifier l'utilisateur et le groupe propriétaire

```
root@deb:~# chown penthium:users /data
```

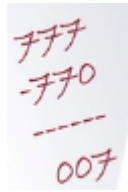
- Modifier seulement le groupe propriétaire de /data/commun ET son contenu

```
root@deb:~# chown -R :users /data/commun
```

## Modèle de droits et droits spéciaux

### Gestion du modèle de droit « Umask »

- L'**umask** est utilisé pour déterminer les droits des fichiers et répertoires.
- La valeur de l'**umask** sera « soustraite » à la valeur des droits maximaux à la création : **0666** pour un fichier, **0777** pour un répertoire (*le 0 est lié aux droits spéciaux*).
- Sur Debian, l'umask par défaut des utilisateurs est de **0022**. Il est important que l'umask de root et des utilisateurs de service (**UID < 1000**) reste à **0022**.
- Il est possible pour chaque utilisateur de changer son umask avec la commande **umask**.
- En pratique, on calcule toujours l'umask à partir des droits pour les répertoires (sinon le droit d'exécution ne sera pas posé pour les dossiers créés avec ce masque.)
- Si je souhaite obtenir des droits de lecture-écriture-exécution pour l'utilisateur et le groupe propriétaire :
  - Je vais calculer la somme en octal : **770**
  - Ensuite je vais soustraire cette valeur cible de **777**, le maximum possible



- Ce qui amène ici le résultat **007**
- Ou **0007** avec prise en compte du premier chiffre représentant les droits spéciaux

```
umask [mode]
```

- Afficher le masque courant

```
user30@deb:~$ umask
0022
```

- Modifier le masque courant

```
user30@deb:~$ umask 0007
user30@deb:~$ umask
0007
```

- Pour avoir un umask spécifique lors de sa connexion, il sera possible d'ajouter une entrée avec la commande **umask** dans son fichier de chargement de son shell (le fichier **~/.bashrc** pour le shell **bash**)

## Les droits spéciaux Unix

Droit	Octal	Fichier	Répertoire
<b>SetUID</b> Symbolique sur colonne user, champ x: s = SetUID + droit x positionné S = SetUID positionné seul	4---	Exécution avec les droits du propriétaire du fichier  Ex : la commande <b>/usr/bin/passwd</b> s'exécute avec les droits de <b>root</b> , afin de permettre l'écriture dans les fichiers <b>passwd</b> et <b>shadow</b>	<i>Non utilisé pour les dossiers</i>
<b>SetGID</b> Symbolique sur colonne group, champ x: s = SetGID + droit x positionné S = SetGID positionné seul	2---	Exécution avec les droits du groupe Ex : <b>/usr/sbin/crontab</b> s'exécute avec les droits du groupe <b>crontab</b>	Tout fichier créé dans ce dossier héritera du groupe du dossier parent. Les sous-dossiers créés hériteront quant à eux du droit SetGID.

Droit	Octal	Fichier	Répertoire
<b>Sticky bit</b> Symbolique sur colonne other, champ x: t = Sticky bit + droit x positionné T = Sticky bit positionné seul	1---	Mise en zone de swap. Le but est que le fichier puisse être stocké en mémoire pour pouvoir être relancé plus rapidement	Tout fichier créé ne pourra être supprimé que par son propriétaire et par root  Ex : <b>/tmp</b>

```
chmod <-R> [ugoa][+ - =][st] fichier ou dossier
```

## Conclusion

- Vous comprenez le fonctionnement des droits Unix Linux
- Vous savez modifier les droits et propriétaires de dossiers et fichiers
- Vous savez manipuler les modèles de droits
- Vous savez positionner des droits spéciaux