

# Droits sur les fichiers et répertoires

---

Les droits des fichiers sont directement inscrits dans l'inode des fichiers. Les droits sont symbolisés par les lettres: **r (read)** ; **w (write)** ; **x (execute)**, l'absence de droit est indiquée avec le symbole -. Pour les consulter, il suffit d'utiliser la commande **ls**:

Par exemple, si nous souhaitons afficher les droits liés au répertoire **Documents**:

## Input:

```
ls -ld Documents
```

## Output:

```
drwxr-xr-x 2 [utilisateur] [NomDuGroupe] [DateEtHeure] [NomDuFichier]
```

## Comprendre les droits

Reprenons l'exemple précédent (output): **drwxr-xr-x**. La lettre **d** indique le type d'objet, ici, un répertoire (directory), les trois lettres suivantes affichent les droits des utilisateurs, les trois suivantes, les droits du groupe et les trois dernières, les droits des autres utilisateurs.

Type d'objet	Utilisateur	Groupe	Autres
d (directory)	rwX	r-X	r-X

## Les droits en mode octal

Chacun de ces droits est aussi égal à une valeur octale:

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

Dans le cas de notre répertoire **Documents**, la valeur de ses droits en mode octal est égale à 755:

Type d'objet	Utilisateur	Groupe	Autres
d	rwX	r-X	r-X
d	4 + 2 + 1	4 + 0 + 1	4 + 0 + 1
d	7	5	5

## Umask

L'umask est un filtre qui définit les droits par défaut des fichiers et des répertoires sous forme octale. Il est possible de le modifier. Pour cela, il faut soustraire à la valeur octale maximale du fichier/répertoire, la valeur de l'umask pour obtenir la valeur octale des permissions souhaitées.

Pour obtenir la valeur octale maximale d'un fichier(666)/répertoire(777), il faut traduire les valeurs octales en binaire. On indique donc dans un tableau nos valeurs octales correspondantes au(x) droit(s) et leur valeur en binaire:

<b>Droit(s)</b>	<b>Valeur octale</b>	<b>Valeur binaire</b>
-	0	000
-x-	1	001
-w-	2	010
-wx	3	011
r--	4	100
r-x	5	101
rw-	6	110
<b>rwX</b>	<b>7</b>	<b>111</b>

Pour déterminer la valeur octale, il suffit de l'établir dans un tableau binaire:

<b>128</b>	<b>64</b>	<b>32</b>	<b>16</b>	<b>8</b>	<b>4</b>	<b>2</b>	<b>1</b>
0	0	0	0	0	1	1	1

Il est maintenant possible d'établir la valeur octale maximale atteignable pour un fichier et pour un répertoire ; la valeur maximale n'est pas la même dans le cas d'un répertoire et d'un fichier, car Linux n'autorise pas l'exécution des fichiers par défaut pour des raisons évidentes de sécurité, la valeur maximale d'un fichier sera donc inférieure à celle d'un répertoire (car le **x** n'est pas attribuable par défaut).

Pour un fichier, la valeur maximale est donc:

```
rw-rw-rw- | soit 666
```

Pour un répertoire:

```
rwXrwxrwx | soit 777
```

Il ne reste maintenant plus qu'à soustraire à ces valeurs maximales, la valeur de notre umask qui déterminera la valeur des nouveaux droits par défaut.

Si nous souhaitons que les répertoires disposent des droits suivants par défaut:

```
r-x-r-x--- | soit 550
```

Il faut taper la commande suivante:

```
umask 0227 | car 777 - 227 = 550
```

Les nouveaux droits par défaut correspondront donc bien à:

```
r-x = 5 r-x = 5 --- = 0 | soit 550
```

Pour vérifier, entrez votre **umask** et créez un fichier:

```
sudo mkdir [NomDuRépertoire]
```

Puis affichez les droits:

```
ls -ld [NomDuRépertoire]
```

Les fichiers, eux, auront les droits suivants:

```
r--r----- | soit 440
```

Car il n'est pas possible d'attribuer le droit **x** par défaut aux fichiers.

**N.B:** L'umask s'applique uniquement sur les fichiers créés après la saisie de celui-ci, les fichiers créés avant ne verront pas leurs droits modifiés.

## Les commandes d'administration des fichiers/répertoires

Pour gérer les droits et les propriétaires des fichiers/répertoires, il existe des commandes:

- **chmod**: permet de modifier les droits liés à un fichier/répertoire.
- **chown**: permet de modifier le propriétaire et/ou le groupe propriétaire d'un fichier/répertoire.

chmod

**syntaxe:**

```
sudo chmod [u,g,o][+/-][r,w,x] [NomDuFichier]
```

Les options **u, g et o** correspondent aux trois catégories d'utilisateurs: **u (utilisateur), g (groupe) et o (autre)** (c.f: premier tableau). Les symboles **+** et **-** indiquent l'ajout ou la suppression de droit(s) et les lettres **r,w,x** correspondent aux droits.

Si l'on souhaite par exemple donner les droits suivants liés au fichier **Test**: **rw-rw----**, avec des valeurs absolues:

On vérifie les droits liés au fichier:

#### Input:

```
ls -l Test
```

#### Output:

```
rw-r--r--
```

Le fichier permet actuellement la lecture et l'écriture pour l'utilisateur propriétaire et la lecture pour le groupe et les autres utilisateurs.

Pour modifier les droits:

```
sudo chmod g+w,o-r Test
```

On indique ici que l'on souhaite ajouter (+) le droit d'écriture au groupe propriétaire (**g**) et supprimer (-) le droit de lecture des autres utilisateurs (**o**) ; il faut séparer les différentes options par une virgule.

Il suffit maintenant de vérifier les droits attribués:

```
ls -l Test
```

Pour effectuer la même attribution de droits avec le mode octal:

```
sudo chmod 640 Test
```

chown

#### Syntaxe:

```
sudo chown [options] [user]:[group] [fichier/répertoire]
```

## Droits spéciaux

Il existe des droits spéciaux liés au système UNIX.

**SetUID** permet à un utilisateur d'exécuter un fichier/répertoire avec les droits du propriétaire dudit fichier, c'est une élévation de privilèges. Le bit setUID se distingue par la lettre **S** qui se substitue à la lettre **x** dans les droits du fichier/répertoire dans la partie utilisateur.

Pour activer le bit setGUI:

```
sudo chmod u+s [NomDuFichier]
```

Pour vérifier l'application du bit:

```
ls -l [NomDuFichier]
```

La lettre **s** devrait être visible dans la partie **utilisateur** et attester de l'application du bit.

Pour supprimer le bit setUID:

```
sudo chmod s-u [NomDuFichier]
```

**SetGID** permet d'exécuter un fichier/répertoire avec les droits du groupe propriétaire. Si le setGID est affecté à un répertoire, il le sera aussi aux fichiers contenus dans celui-ci. Le bit setGID se distingue par la lettre **S** qui se substitue à la lettre **x** dans les droits du fichier/répertoire dans la partie groupe.

Pour appliquer le bit setGID:

```
sudo chmod g+s [NomDuFichier]
```

Pour vérifier l'application du bit:

```
ls -l [NomDuFichier]
```

La lettre **S** devrait être visible dans la partie **groupe** et attester de l'application du bit.

Pour supprimer le bit setGID:

```
sudo chmod g-s [NomDuFichier]
```

---

**Stickybit**, lorsqu'il est appliqué à un fichier/répertoire, ce dernier ne peut être modifié/supprimé que par l'utilisateur propriétaire, le groupe propriétaire ou le super-utilisateur (root). Le stickybit est discernable par la lettre **T** dans les droits du fichier/répertoire.

Pour appliquer le stickybit:

```
sudo chmod +t [NomDuFichier]
```

Pour vérifier l'application du bit:

```
ls -l [NomDuFichier]
```

La lettre **T** devrait être visible à la fin des droits et attester de l'application du bit.

Pour supprimer le stickybit:

```
sudo chmod -t [NomDuFichier]
```

**N.B:** À noter que les droits spéciaux distinguent les droits **s** et **t**, qui indiquent le **droit d'exécution** et le **bit spécial**. Alors que les droits **T** et **S**, eux, indiquent **uniquement** la pose du **droit spécial**.