

# Les droits d'accès

## Identification de l'utilisateur

On a vu auparavant que pour pouvoir se connecter sur une machine, on doit être déclaré sur la machine. Tout utilisateur appartient à un groupe, concrètement dans une université par exemple vous aurez les professeurs dans le groupe enseignant et les élèves dans le groupe élève.

Chaque utilisateur est identifié par un numéro unique UID (User identification), de même chaque groupe est identifié par un numéro unique GID (Group identification).

Vous pouvez voir votre UID et GID en éditant le fichier `/etc/passwd`, c'est respectivement troisième et quatrième champ, après le nom (le login), et le mot de passe crypté.

## Définition des droits d'utilisateur

### Cas d'un fichier classique

Avec UNIX les fichiers bénéficient d'une protection en lecture, écriture et exécution, c'est à dire vous pouvez choisir si vous voulez que vos fichiers soient lisibles et/ou modifiables par d'autres, vous pouvez empêcher que d'autres utilisateurs lancent vos exécutables. C'est le principe des droits d'accès.

Nous avons vu qu'en tapant `ls -l` le premier champ correspondait au droit d'accès, on avait une sortie de ce type :

```
-rwxrw-r-- 1 olivier users 34568 Dec 3 14 :34 mon-fichier
```

La signification des lettres `rwX` et la suivante :

`r` (read) on peut lire le fichier `w` (write) on peut modifier le fichier `x` (exécutable) on peut exécuter le fichier (c'est donc un exécutable) - aucun droit autorisé

Le champ `-rwxrw-r--` regroupe les droits du propriétaire du fichier, du groupe auquel appartient le propriétaire et les autres utilisateurs.

- on a affaire à un fichier classique (c'est à ni un répertoire, ni un fichier spécial) `rwX` droits sur le fichier du propriétaire `rw-` droits sur le fichier du groupe auquel appartient le propriétaire (users) `r--` droits sur le fichier des autres utilisateurs (ceux n'appartenant au groupe users)

Par exemple pour notre fichier le propriétaire `olivier` a des droits en écriture, lecture et exécution, le groupe a un droit en lecture et écriture mais aucun droit en exécution, les autres utilisateurs ont uniquement le droit en lecture du fichier.

Pour info le **1** après les droits signifie que le fichier **mon-fichier** n'a aucun lien qui pointe vers lui, si on avait eu **2**, cela signifiait que quelque part dans l'arborescence, il y a un lien qui pointe vers lui, ce nombre s'incrémentant avec le nombre de lien.

## Cas d'un répertoire

Pour un répertoire le **x** n'est pas un droit en exécution, mais un droit d'accès au répertoire, sans ce droit, on ne peut pas accéder au répertoire et voir ce qu'il y a dedans.

En tapant **ls -l** sur un répertoire, vous obtenez :

```
drwxr-x--- 1 olivier users 13242 Dec 2 13 :14 mon-répertoire
```

**d** signifie qu'on a affaire à un répertoire, **rw**x sont les droits du propriétaire **olivier** qui est autorisé en lecture, écriture et droit d'accès au répertoire, **-x** droits du groupe **users**, autorisé en lecture, droit d'accès au répertoire, pas de droit en écriture — droits des autres utilisateurs, aucun droit dans le cas présent

## Cas d'un lien

Pour un lien, la signification est similaire à celle d'un fichier classique, à la différence que vous avez un **l** à la place du **-** en tout début de ligne.

```
lrwxrwxrwx 1 root root 14 Aug 1 01:58 Mail -> ../../bin/mail*
```

[\[Retour haut de la page\]](#)

## Commandes associées

### Changer les droits : **chmod**

La commande **chmod** permet de modifier les droits d'accès d'un fichier (ou répertoire). Pour pouvoir l'utiliser sur un fichier ou un répertoire, il faut en être le propriétaire. La syntaxe est la suivante :

chmod utilisateur	opération	droit d'accès
u propriétaire (user)	+ajout d'un droit	r droit en lecture
g groupe (group)	-suppression d'un droit	w droit en écriture
o les autres (other)	=ne rien faire	x*

\* droit en exécution pour un fichier, droit d'accès pour un répertoire.

Exemple vous voulez donner un droit en écriture pour le groupe du fichier **mon-fichier**

```
chmod g+w mon-fichier
```

Pour supprimer le droit d'accès du répertoire **mon-répertoire** aux autres utilisateurs (autres que propriétaire et utilisateurs du groupe)

```
chmod o-x mon-repertoire
```

En tapant

```
chmod u+x,g-w mon-fichier
```

Vous ajoutez le droit en exécution pour le propriétaire, et enlevez le droit en écriture pour le groupe du fichier.

Vous avez une autre méthode pour vous servir de la commande **chmod**. On considère que  $r=4$ ,  $w=2$  et  $x=1$ , si vous avez un fichier avec les droits suivants **-rw-rw-rw-**, pour les droits utilisateurs vous avez  $(r=)4+(w=)2=6$ , de même pour le groupe et les autres. Donc **-rw-rw-rw-** est équivalent à **666**. En suivant la même règle **rw-rw-r-** est équivalent à **754**.

Pour mettre un fichier avec les droits **-r--r--** vous pouvez taper :

```
chmod 444 mon-fichier
```

On appelle ce système de notation, la notation octale.

## Changer les droits par défaut : umask

Quand vous créer un fichier, par exemple avec la commande **touch**, ce fichier par défaut possède certains droits. Ce sont **666** pour un fichier (**-rw-rw-rw-**) et **777** pour un répertoire (**-rwxrwxrwx**), ce sont les droits maximum. Vous pouvez faire en sorte de changer ces paramètres par défaut. La commande **umask** est là pour ça.

Pour un fichier :

Si vous tapez **umask 022**, vous partez des droits maximum **666** et vous retranchez **022**, on obtient donc **644**, par défaut les fichiers auront comme droit **644** (**-rw-r--**).

Si vous tapez **umask 244**, vous partez des droits maximum **666** et vous retranchez **244**, on obtient donc **422**, par défaut les fichiers auront comme droit **422** (**-rw--w--**).

Pour un répertoire :

Si vous tapez **umask 022**, vous partez des droits maximum **777** et vous retranchez **022**, on obtient donc **755**, par défaut les fichiers auront comme droit **644** (**-rwxr-xr-x**).

Si vous tapez **umask 244**, vous partez des droits maximum **777** et vous retranchez **244**, on obtient donc **533**, par défaut les fichiers auront comme droit **422** (**-rwx-wx-wx**).

**umask** n'est utilisable que si on est propriétaire du fichier.

## Changer le propriétaire et le groupe

Vous pouvez " donner " un fichier vous appartenant à un autre utilisateur, c'est à dire qu'il deviendra propriétaire du fichier, et que vous n'aurez plus que les droits que le nouveau propriétaire voudra bien vous donner sur le fichier.

```
chown nouveau-propriétaire nom-fichier
```

Dans le même ordre d'idée vous pouvez changer le groupe.

```
chgrp nouveau-groupe nom-fichier
```

Ces deux commandes ne sont utilisables que si on est propriétaire du fichier.

**NOTA** : Sur certains UNIX suivant leur configuration, on peut interdire l'usage de ces commandes pour des raisons de sécurité.

[\[Retour haut de la page\]](#)