

# Les commandes Linux (3<sup>e</sup> partie) : fichiers, répertoires et droits d'accès

Sous LINUX chaque fichier et chaque répertoire est régi par des droits visant à protéger l'ensemble du système.

## 1. Utilisateurs et droits.

### 1.1. Catégorie d'utilisateurs.

Pour chaque fichier ou répertoire il y a 3 catégories d'utilisateurs : l'utilisateur normal, c'est le propriétaire, bien souvent son créateur : **u (user)**, les membres du même groupe : **g (group)**, tous les autres : **o (others)**.

### 1.2. Droits.

Chaque fichier ou répertoire est doté d'une structure appelé i-node où sont stockées les droits sous la forme de trois bits d'accès (lecture **r (read)**, écriture **w (write)** et exécution **x (execute)**) pour chacune des catégories d'utilisateurs. Ces droits n'ont, toutefois, pas la même signification lorsqu'ils s'appliquent à un fichier ou à un répertoire

	pour un fichier	pour un répertoire
<b>read</b>	lecture autorisée	Droit de lire le contenu, d'afficher la liste des fichiers (avec ls)
<b>write</b>	modification autorisée	Modification du contenu : droits d'ajouter et de supprimer des fichiers (avec cp, mv, rm)
<b>execute</b>	exécutable	Droit d'accéder aux fichiers du répertoire et de se déplacer à l'intérieur (avec cd)

**Note** : si on attribue **w**, il faut attribuer aussi **x** sur un répertoire.

Les droits se combinent en 3 blocs rwx. Le système de droits est spécifié symboliquement par 9 attributs, correspondants aux 3 catégories d'utilisateurs du fichier.

... | ... | ...  
u g o

En cas d'absence de droit la lettre est remplacée par un tiret. Ces droits sont consultables par la commande «ls -l». En plus des droits cette commande affiche un autre caractère : "-" pour un fichier, "d" pour un répertoire, "l" pour un lien symbolique.

Exemple :

```
-rwxr--r-- 1 root root 845 oct 12 20:14 taille.sh*
-rwxr-xr-x 1 root root 1488 oct 12 20:14 themes.sh*
drwxr-xr-x 2 root root 1024 mai 25 19:17 tmp/
drwxr-xr-x 6 root root 1024 oct 16 18:35 travaux/
lrwxrwxrwx 1 root root 20 oct 21 21:40 vcron -> /usr/local/bin/vcron
```

A la première ligne, il s'agit d'un fichier sur lequel l'utilisateur a le droit de lecture, d'écriture et d'exécution. Les membres du groupe ont seulement le droit de lecture. Il en est de même pour les autres.

A la seconde ligne les membres du groupe et les autres ont en plus le droit d'exécution.

Les lignes 3 et 4 désignent 2 répertoires.

La cinquième ligne désigne un lien vers le fichier /usr/local/bin/vcron sur lequel tout le monde a tous les droits.

La commande «stat» fournit des informations plus complètes :

```
[root@linux /root]# ls -l taille.sh
-rwxr--r-- 1 root root 845 oct 12 20:14 taille.sh*
[root@linux /root]# stat taille.sh
File: "taille.sh"
Size: 845 Allocated Blocks: 2 Filetype: Regular File
Mode: (0744/-rwxr--r--) Uid: ( 0/ root) Gid: ( 0/ root)
Device: 3,1 Inode: 41502 Links: 1
Access: Thu Oct 12 20:14:09 2000
Modify: Thu Oct 12 20:14:09 2000
Change: Thu Oct 12 20:14:09 2000
```

### 1.3. La notation octale.

La commande «stat» nous montre une autre manière de marquer les droits (voir ci-dessus) en utilisant la notation octale. Par convention la présence d'un droit est noté «1», l'absence «0». On a la correspondance suivante :

Binaire	-----	Droit	-----	Octal
000	-----	(---)	-----	0
001	-----	(--x)	-----	1
010	-----	(-w-)	-----	2
011	-----	(-wx)	-----	3
100	-----	(r--)	-----	4
101	-----	(r-x)	-----	5
110	-----	(rw-)	-----	6
111	-----	(rwx)	-----	7

Il est habituel de représenter les droits en octal comme indiqué dans le tableau suivant :

propriétaire			groupe			autres		
lecture	écriture	exécution	lecture	écriture	exécution	lecture	écriture	exécution
400	200	100	40	20	10	4	2	1

Pour obtenir les permissions exprimées en octal, il suffit d'ajouter les nombres de la table de correspondance ci-dessus, pour lesquels les droits sont positionnés.

Dans l'exemple du fichier «*taille.sh*» vu précédemment, on a :

pour le propriétaire : rwx donc  $400 + 200 + 100 = 700$

pour le groupe : r-- donc  $40 + 0 + 0 = 40$

pour les autres : r-- donc  $4 + 0 + 0 = 4$

total = **744**

## 2. Les commandes pour la gestion des droits.

Trois commandes servent à modifier les droits : `chown` (change owner), `chgrp` (change group) et `chmod` (change mode). Elles sont utilisables par root ou le propriétaire.

**chown** : changer le propriétaire d'un fichier, d'un répertoire ou d'un ensemble de ces éléments.

Syntaxe : `chown [options] nouvel_utilisateur [fichier(s)] [répertoire(s)]`

**chgrp** : changer le groupe propriétaire.

Utilisable par root ou le propriétaire, à condition que celui-ci soit membre du nouveau groupe.

Syntaxe : `chgrp [options] nouveau_groupe [fichier(s)] [répertoire(s)]`

La commande `chgrp` est redondante car `chown` permet de changer le groupe seul, ainsi que le propriétaire et le groupe en même temps.

Pour les deux commandes l'option «-R» permet de changer propriétaire et/ou groupe récursivement.

Dans l'exemple ci-dessous phil n'est pas autorisé à modifier le propriétaire.

```
[phil@linux phil]$ ls -l
drwxrwxr-x 2 phil phil 1024 oct 22 10:13 essai1/
-rw-rw-r-- 1 phil phil 8 oct 22 10:13 essai2
[phil@linux phil]$ chown root essai1 essai2
chown: essai1: Opération non permise
chown: essai2: Opération non permise
```

En tant que root c'est possible.

```
[root@linux /root]# cd /home/phil
[root@linux phil]# chown root essai1 essai2
[root@linux phil]# ls -l
drwxrwxr-x 2 root phil 1024 oct 22 10:13 essai1/
-rw-rw-r-- 1 root phil 8 oct 22 10:13 essai2
```

Changements de groupes avec les 2 commandes.

```
[root@linux phil]# chgrp root essai1
[root@linux phil]# chown .root essai2
[root@linux phil]# ls -l
drwxrwxr-x 2 root root 1024 oct 22 10:13 essai1/
-rw-rw-r-- 1 root root 8 oct 22 10:13 essai2
```

Changements de groupe et de propriétaire simultanément :

```
[root@linux phil]# chown phil.phil essai2
[root@linux phil]# ls -l
drwxrwxr-x 2 root root 1024 oct 22 10:13 essai1/
-rw-rw-r-- 1 phil phil 8 oct 22 10:13 essai2
```

Changements récursifs :

```
[root@linux phil]# ls -l essai1
-rw-r--r-- 1 root root 42 oct 23 14:33 fichier1.txt
-rw-r--r-- 1 root root 275 oct 23 14:33 fichier2.txt
[root@linux phil]# chown -R phil.phil essai1
[root@linux phil]# ls -l
drwxrwxr-x 2 phil phil 1024 oct 23 14:33 essai1/
-rw-rw-r-- 1 phil phil 8 oct 22 10:13 essai2
[root@linux phil]# ls -l essai1
-rw-r--r-- 1 phil phil 42 oct 23 14:33 fichier1.txt
-rw-r--r-- 1 phil phil 275 oct 23 14:33 fichier2.txt
```

**chmod** : changer les permissions sur les fichiers ou répertoires (effectué par root ou le propriétaire)

syntaxe : `chmod [options] droits [fichier(s)] [répertoire(s)]`

La commande «`chmod`» peut s'écrire de plusieurs manières. Les droits peuvent se définir de façon relative, par ajout ou retrait par rapports aux droits existants ou bien de façon absolue, en remplaçant les anciens droits par les nouveaux droits.

### Notation relative (aux droits existants).

Syntaxe : `chmod [options] [ugoa] [+|=] [rwx] [fichier(s)] [répertoire(s)]`

où «u, g et o» désignent les 3 catégories d'utilisateurs et «a» (=all) tous, «r,w,x» les attributs, «+, -, =» l'action d'ajouter, de retirer ou de fixer un droit s'appliquant à une catégorie.

Exemples :

```
[root@linux phil]# ls -l
drwx----- 5 phil phil 1024 oct 6 19:24 Desktop/
...
[root@linux phil]# chmod go+rwx Desktop
[root@linux phil]# ls -l
drwxrwxrwx 5 phil phil 1024 oct 6 19:24 Desktop/
[root@linux phil]# ls -l essai2
-rw-rw-r-- 1 phil phil 8 oct 22 10:13 essai2
[root@linux phil]# chmod go-r essai2
[root@linux phil]# ls -l essai2
-rw--w---- 1 phil phil 8 oct 22 10:13 essai2
```

### Notation absolue.

Syntaxe : `chmod [options] u=...,g=...,o=... [fichier(s)] [répertoire(s)]`

Pour chaque catégorie d'utilisateur, elle permet de fixer les nouveaux droits qui remplacent les anciens. Si une catégorie n'est pas présente, ses anciens droits s'appliquent.

L'option «-R» permet de changer les droits récursivement

```
[root@linux phil]# ls -l
drwxr--r-- 5 phil phil 1024 oct 6 19:24 Desktop/
[root@linux phil]# chmod g=r,o=r Desktop
[root@linux phil]# ls -l
drwxr--r-x 5 phil phil 1024 oct 6 19:24 Desktop/
[root@linux phil]# chmod g=r,o=rwt Desktop
[root@linux phil]# ls -l
drwxr--rwt 5 phil phil 1024 oct 6 19:24 Desktop/
[root@linux phil]# ls -l
drwxrwxr-t 2 phil phil 1024 oct 23 14:33 essai1/
[root@linux phil]# ls -l essai1
total 2
-rwSr--r-- 1 phil phil 42 oct 23 14:33 fichier1.txt
-rw-r-sr-x 1 phil phil 275 oct 23 14:33 fichier2.txt*
[root@linux phil]# chmod -R u=rwx,g=r,o= essai1
[root@linux phil]# ls -l
drwxr----- 2 phil phil 1024 oct 23 14:33 essai1/
[root@linux phil]# ls -l essai1
total 2
-rwxr----- 1 phil phil 42 oct 23 14:33 fichier1.txt*
-rwxr----- 1 phil phil 275 oct 23 14:33 fichier2.txt*
```

## 3. Les outils graphiques.

Sous debian, le gestionnaire de fichiers permet de gérer l'ensemble des droits sur les fichiers et répertoires.

