

Manipulation des fichiers : changement de droits

Toute ressource (fichier, répertoire, ...) a :

- un identificateur
- un propriétaire
- un ensemble de droits d'accès (en lecture, en écriture, en exécution) :
 - les droits du propriétaire
 - les droits du groupe auquel appartient le propriétaire
 - les droits des autres utilisateurs
- Root, administrateur

Manipulation des fichiers : changement de droits

- **chmod** [options] droits fichier1, fichier2, ...
changer les droits d'accès au fichier.

Les droits peuvent être spécifiés de **deux façons**, avec des lettres ou avec des nombres en Octal.

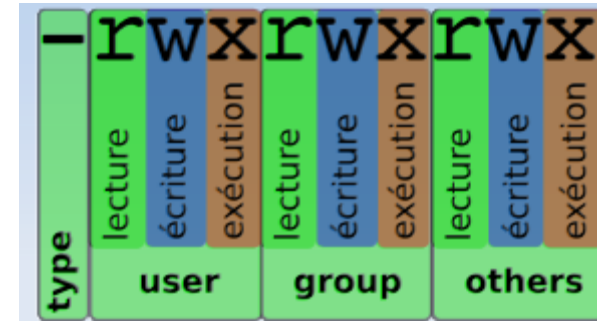
- Pour les lettres, il existe les opérateurs de changement d'état + et - pour ajouter ou retirer un type de droit aux droits courants, et l'opérateur = pour les écraser.
- Pour l'octal, il faut additionner les nombres pour chaque type de possesseur.

Manipulation des fichiers : changement de droits (**Première Méthode**)

- **chmod** [options] droits fichier1, fichier2, ...

Changer les droits d'accès au fichier. Les droits sont définis comme suit :

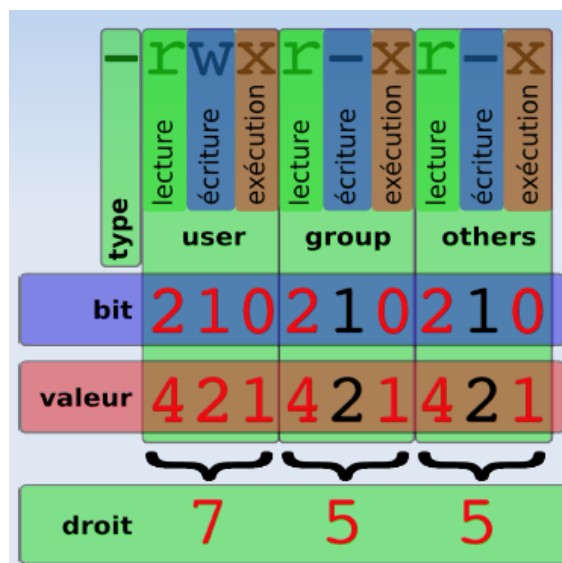
- **u** droits de l'utilisateur (user)
- **g** droits des utilisateurs du groupe (group)
- **a** droits de tous les utilisateurs (all)
- **+r** droit en lecture accordé **-r** droit en lecture retiré
- **+w** droit en écriture accordé **+x** droit d'exécution accordé



Type		Droit		Destinataire
b	Block device	r	Read	u User (propriétaire)
c	Character device file	w	Write	g Groupe
d	Répertoire (Directory)	x	Execute (fichier)	o Others (les autres, ni 'u' ni 'g')
l	Lien symbolique	x	chdir (répertoire)	
s	Socket	s	SUID bit	
p	FIFO			
-	Fichier normal			

Manipulation des fichiers : changement de droits (**Deuxième méthode**)

- Dans la deuxième façon de faire, les permissions sont (valeurs octales entre parenthèses): **r (4)** : autorisation de lecture **w (2)** : autorisation d'écriture **x (1)** : autorisation d'exécution



Correspondances de représentation des droits		
Droit	Valeur alphanumérique	Valeur octale
aucun droit	---	0
exécution seulement	--x	1
écriture seulement	-w-	2
écriture et exécution	-wx	3
lecture seulement	r--	4
lecture et exécution	r-x	5
lecture et écriture	rw-	6
tous les droits (lecture, écriture et exécution)	rwX	7

Manipulation des fichiers : changement de droits

Exemples:

- `chmod a+r fichier` → tout (a) le monde a le droit de lire.
- `chmod 444 fichier` → idem
- `chmod u+x fichier` → l'utilisateur a le droit d'exécuter.
- `chmod 744 fichier` → L'utilisateur a tous les droits, le groupe et les autres ne peuvent que consulter le fichier
- `chmod 755 mon_dossier` → donne au propriétaire tous les droits, aux membres du groupe et aux autres les droits de lecture et d'accès. C'est un droit utilisé traditionnellement sur les répertoires.
- `chmod og-w fichier` → les autres (o) et le groupe (g) n'ont pas le droit d'écrire.

Manipulation des fichiers : commandes d'administration

chown [options] utilisateur.groupe fichier

change le propriétaire d'un fichier

NB: nécessite d'être administrateur (super-user)

Exemple :

```
sudo chown -R paul.L1 projet/
```

Définit l'utilisateur paul et le groupe L1 au répertoire projet et à tous ses fichiers

useradd [options] login

Ajouter un utilisateur

passwd [options] login

Changer de mot de passe

Manipulation des fichiers : copie

- **cp** [-ipr] source dest
- **cp** [option] [chemin vers fichier source] [chemin vers fichier destination]
- **Source** = **fichier**
- **Dest** = **fichier** : copie un fichier source en le renommant si le chemin du fichier destination contient un nom de fichier
- **Dest** = **répertoire** : recopier dans dest

options

- ❖ **-i** : confirmation en cas d'écrasement
- ❖ **-p** : préserve les attribus (propriétaire, groupe, date de création)
- ❖ **-r** : copie récursive (pour les répertoires imbriqués)

Exemples :

\$ cp toto /tmp/ → copie le fichier local toto dans /tmp (toujours nommé toto)
\$ cp toto /tmp/tata → copie le fichier local toto dans /tmp en le nommant tata
\$ cp -r projet /tmp → copie le contenu du répertoire projet dans le répertoire /tmp/projet

Manipulation des fichiers : Déplacement

mv [option] [chemin vers fichier source] [chemin vers fichier destination]

Déplace un fichier source en le renommant si le chemin du fichier destination contient un nom de fichier

Exemples :

\$ mv toto /tmp/ → déplace le fichier local toto dans /tmp (toujours nommé toto)

\$ mv toto /tmp/tata → déplace le fichier local toto dans /tmp en le nommant tata

\$ mv -i toto /tmp → déplace le fichier toto dans /tmp en prévenant l'utilisateur s'il existe déjà un fichier /tmp/toto

Manipulation des fichiers : Suppression

rm [option] [chemin vers fichier]

supprime un fichier

Exemples :

\$ rm toto → supprime le fichier toto

\$ rm -i toto → supprime le fichier toto en demandant confirmation à l'utilisateur

\$ rm -f toto* → supprime les fichiers dont le nom commence par toto, sans demander confirmation à l'utilisateur

\$ rm -r projet → efface récursivement le contenu du répertoire projet **non vide**

Manipulation des fichiers Créer / supprimer un répertoire

mkdir [chemin vers répertoire]

créer un répertoire

rmdir [chemin vers répertoire]

Supprimer un répertoire **vide**

Sécurité: ne fonctionne que quand les répertoires sont vides

Alternative: **rm -r**

Exemples :

\$ mkdir toto → crée le répertoire toto

\$ rmdir toto → supprime le répertoire vide toto

\$ rmdir projet → rmdir: projet/: Directory not empty

Recherche

- **find** [options]

chercher dans une hiérarchie de répertoires les fichiers vérifiant certains caractéristiques données en options.

Exemples

- **find** / -name charte -print

chercher à partir de la racine tous les fichiers dont le nom est charte.

- **find** . -name *.kwd -print

chercher à partir du répertoire courant tous les fichiers dont l'extension est .kwd

- **which** commande

effectue une recherche dans la liste des exécutables de la commande donnée et retourne le chemin d'accès complet d'une commande.

Ex. **which echo** ! → **bin/echo**

Commandes diverses

- **who**

lister des utilisateurs connectés au système

- **date**

afficher date et heure

- **file** fichier

déterminer le type du fichier

- **head** [-n] fichier

afficher les n premiers lignes du fichier

- **tail** [+n|-n] fichier

+n : afficher à partir de la ligne numéro n

-n : afficher le n dernières lignes

- **more** fichier

afficher le fichier page par page

- **sort** fichier

Trier le contenu d'un fichier

- **wc** [-cwl] fichier

-c : nombre de caractères

-w : nombre de mots

-l : nombre de lignes

Commandes diverses

Connaitre l'espace occupé par un répertoire / disque ?

du [option] fichier

Donne la taille en octets d'un fichier

df [option]

Donne la taille des données présentes sur chaque disque

Exemples :

du -sh projet → 4.0K projet/

Manipulation d'archives et compressions

tar cf projet.tar projet/*

crée une archive contenant le contenu du répertoire projet et nommée projet.tar

tar xf projet.tar

extraît le contenu de l'archive nommée projet.tar

tar zcf projet.tar projet/*

crée et compresse une archive contenant le contenu du répertoire projet et nommée projet.tar.gz

tar zxf projet.tar.gz

extraît le contenu de l'archive compressée nommée projet.tar.gz
options

- **-c** : créer une archive
- **-x** : extraire les fichiers
- **-v** : obtenir une description du contenu archivé
- **-f** : pour spécifier un nom pour l'archive (en paramètre)
- **-z** : zipper/dézipper l'archive

Manipulation d'archives et compressions

gzip fichier

Comprime un fichier au format .gz (algorithme deflate)

gunzip fichier

Décompresse un fichier au format .gz

Exemples :

gzip toto.txt → toto.txt.gz

gunzip toto.txt.gz → toto.txt

Manipulation de texte

- **grep** [options] motif fichier1 fichier2 .. (**motif= expression régulière**)

Chercher les lignes contenant le motif décrit par l'expression régulière dans les fichiers

options

- **-c** : indique seulement le nombre de lignes
- **-n** : indique les numéros des lignes trouvés
- **-i** : ne distingue pas majuscule et minuscules

Exemples :

\$ grep "listeria" /home/Cath/cours/* → cherche, dans les fichiers du répertoire cours, des fichiers contenant le motif listeria

\$ grep -n "listeria" /home/Cath/cours/* → idem, mais en affichant le numéro de ligne

\$ grep -c "listeria" /home/Cath/cours/* → idem, mais en donnant le nombre d'occurrences du motif

- **sort** [-r] fichier

Trier les lignes du fichier ou l'entrée standard.

option

- **-r** : renverser le tri