

Gestion des droits sur Linux

Les droits

Chaque utilisateur possède certains droits lui permettant d'effectuer certaines opérations et pas d'autres. Le système d'exploitation permet de gérer ces droits très finement.

Les fichiers et les répertoires possèdent 3 types de droits :

- les droits en lecture (symbolisés par la lettre r) : est-il possible de lire le contenu de ce fichier
- les droits en écriture (symbolisés par la lettre w) : est-il possible de modifier le contenu de ce fichier
- les droits en exécution (symbolisés par la lettre x) : est-il possible d'exécuter le contenu de ce fichier (quand le fichier du code exécutable)

Il existe 3 types d'utilisateurs pour un fichier ou un répertoire :

- le propriétaire du fichier (par défaut c'est la personne qui a créé le fichier), il est symbolisé par la lettre u
- un fichier est associé à un groupe, tous les utilisateurs appartenant à ce groupe possèdent des droits particuliers sur ce fichier. Le groupe est symbolisé par la lettre g
- tous les autres utilisateurs (ceux qui ne sont pas le propriétaire du fichier et qui n'appartiennent pas au groupe associé au fichier). Ces utilisateurs sont symbolisés la lettre "o"

Lecture des droits

Il est possible d'utiliser la commande "ls" avec l'option "-l" afin d'obtenir des informations supplémentaires détaillées.

Par exemple :

```
-rw-r--r-- 1 william duruy 256 juin 18 19:58 essai.txt  
drwxr-xr-x 2 william duruy 4096 juin 18 20:05 info
```

- le premier symbole "-" `rw-r--r-- 1` signifie que l'on a affaire à un fichier, dans le cas d'un répertoire, nous aurions un "d" (voir la 2e ligne)
- les 3 symboles suivants "rw-" `rw-r--r-- 1` donnent les droits du propriétaire du fichier : lecture autorisée (r), écriture autorisée (w), exécution interdite (- à la place de x)
- les 3 symboles suivants "r--" `rw-r--r-- 1` donnent les droits du groupe lié au fichier : lecture autorisée (r), écriture interdite (- à la place de w), exécution interdite (- à la place de x)
- les 3 symboles suivants "r--" `rw-r--r-- 1` donnent les droits des autres utilisateurs : lecture autorisée (r), écriture interdite (- à la place de w), exécution interdite (- à la place de x)
- le caractère suivant "1" donne le nombre de liens (nous n'étudierons pas cette notion ici)
- "william" représente le nom du propriétaire du fichier
- "duruy" représente le nom du groupe lié au fichier
- le "256" représente la taille du fichier en octet (ici notre fichier est vide)
- "juin 18 19:58" donne la date et l'heure de la dernière modification du fichier
- "essai.txt" est le nom du fichier

Devinette :

Si vous faites partie du groupe duruy, pouvez-vous modifier le fichier `essai.txt`

Modification des droits

Le propriétaire d'un fichier peut modifier les permissions d'un fichier ou d'un répertoire à l'aide de la commande "chmod". Pour utiliser cette commande, il est nécessaire de connaître certains symboles :

- les symboles liés aux utilisateurs : "u" correspond au propriétaire, "g" correspond au groupe lié au fichier (ou au répertoire), "o" correspond aux autres utilisateurs et "a" correspond à "tout le monde" (permet de modifier "u", "g" et "o" en même temps)
- les symboles liés à l'ajout ou la suppression des permissions : "+" on ajoute une permission, "-" on supprime une permission, "=" les permissions sont réinitialisées (permissions par défaut)
- les symboles liés aux permissions : "r" : lecture, "w" : écriture, "x" : exécution.

La commande "chmod" à cette forme :

```
chmod [u g o a] [+ - =] [r w x] nom_du_fichier
```

par exemple :

```
chmod o+w toto.txt
```

attribuera la permission "écriture" pour le fichier "toto.txt" "aux autres utilisateurs"

Exemple :

analysez le code ci-dessous :

```
william@PC ~ $ cd nsi
william@PC ~ / nsi $ ls-l
total 1
-rw-rw-r-x 1 william duruy 256 juin 18 21 :29 toto.txt
william@PC ~ / nsi $ chmod g-w toto.txt
william@PC ~ / nsi $ ls-l
total 1
-rw-rw-r-x 1 william duruy 256 juin 18 21 :30 toto.txt
```

C'est à vous :

Créez un répertoire "test_nsi" dans votre répertoire personnel. Placez-vous dans le répertoire "test_nsi". Créez un fichier "titi.txt", vérifiez les permissions associées à ce fichier. Modifiez les permissions associées au fichier "titi.txt" afin que les "autres utilisateurs" aient la permission "écriture"

Un peu de binaire

Propriétaire			Groupe			Autres		
Lecture	Ecriture	Exécution	Lecture	Ecriture	Exécution	Lecture	Ecriture	Exécution
r	w	x	r	w	x	r	w	x
4	2	1	4	2	1	4	2	1

Les droits sont définis sur un nombre à 3 caractères (3 bits) . Le premier indique les droits du propriétaire du fichier, le deuxième chiffre indique les droits du groupe associé au fichier et le dernier indique le droits de tous les autres utilisateurs. On codera donc chaque triplet en binaire

On ajoute alors pour chaque triplet les droits obtenus, le droit le plus élevé étant 7 puisque $4 + 2 + 1$. Ainsi, on obtient un entier entre 0 et 7 et on parle alors de numérotation Octale.

Toutes les valeurs possibles des droits

Droit	Valeur alphanumérique	Valeur octale	Valeur binaire
Aucun droit	---	0	000
Exécution	--x	1	001
Ecriture	-w-	2	010
Ecriture et exécution	-wx	3	011
Lecture seulement	r--	4	100
Lecture et exécution	r-x	5	101
Lecture et écriture	rw-	6	110
Tous les droits	rwX	7	111

Un petit exemple :

Essayons de comprendre à travers l'exemple cité plus haut.

Quels sont les droits du dossier dont ls-l donne : `drwxr-x--x?`

Découpons l'information :

`drwxr-x--x`

Légende :

Propriétaire	Lecture + Ecriture + Exécution	7
Groupe propriétaire	Lecture + - + Exécution	5
Les autres	Exécution	1

On obtient alors **751**

Utiliser le shell pour effectuer des tâches répétitives.

Outre le fait que manipuler des fichiers au clavier plutôt qu'à la souris est souvent plus rapide, le shell prend toute son utilité lorsqu'il peut nous éviter d'effectuer des tâches répétitives, et donc d'ordinaire très longues et fastidieuses.

Pour cela, on utilise le signe * qui joue le rôle d'un « joker ».

Par exemple :

- **rm toto*** efface tous les fichiers commençant par **toto**.
- **ls toto*** liste tous les fichiers commençant par **toto**.

Situation 1 :

À l'aide de l'interface graphique de gestion de fichiers recopier le dossier toto dans votre répertoire personnel.

1. Dans le répertoire **a_voir** effacer tous les fichiers commençant par **inutiles**.
2. Vider le répertoire **corbeille**.
3. Trouver une commande simple permettant de lister tous les fichiers se trouvant dans le répertoire **~/toto/456679fgt78/asor482fljjlmk**

Situation 2 :

Suite à l'effacement accidentel d'un grand nombre de fichiers, nous avons du faire appel à un logiciel de récupération de fichiers.

Ceux-ci ont été stockés dans le répertoire **~/toto/recup**.

1. Lister la liste des fichiers qui se trouvent dans ce répertoire.

Afin de les réorganiser au mieux, il s'agit de mettre les fichiers se terminant par **mp3** dans un répertoire **~/toto/music**, ceux se terminant par **png** ou **jpeg** dans un répertoire **~/toto/images** et ceux se terminant par **pdf** dans un répertoire **~/toto/documents**.

2. Créer les répertoires **music**, **images** et **documents** dans **toto**

3. Lister tous les fichiers du répertoire **recup** se terminant par **mp3**.

4. Déplacer tous les fichiers de **recup** se terminant par **mp3** dans le répertoire **music** par la commande :

```
mv recup/*mp3 music
```

5. Ranger les fichiers **pdf** et **png** ou **jpg**.

Situation 3 :

Dans le répertoire `~toto/images` se trouvent des images de différentes dimensions.

Il s'agit de modifier la largeur des images **png** afin que celle-ci soit de **200 pixels**.

Il faudrait normalement ouvrir chaque image avec un logiciel comme Gimp ou Genviewer, puis les redimensionner (en conservant l'échelle).

Ça va être long ...

Nous pouvons heureusement utiliser la commande :

```
for fichier in *.png ; do convert $fichier -resize 200x new_$fichier; done
```

1. Placez-vous dans le répertoire `~/toto/images`.
2. Modifier la largeur des images png qui doit être de 200.
3. Choisissez une image png, comment vérifier que l'échelle a bien été respectée ?
4. Modifier la hauteur des images jpg qui doit être de 100.

Exercice :

Enlever tous les droits à group et other pour tous les fichiers se trouvant dans `~/toto/documents`.

Situation 4 :

La commande :

convert image.png image.jpg

permet de convertir le format d'image **png** en **jpg**.

1. Transformer tous les **nouveaux** fichiers png en jpg.
- 2.