

[Fondamentaux] Gestion des droits

Introduction

Linux permet une gestion fine des droits sur les fichiers et les dossiers du système. Cette page ne traitera pas de sudo qui a déjà une page dédiée.

Commandes

ls -l

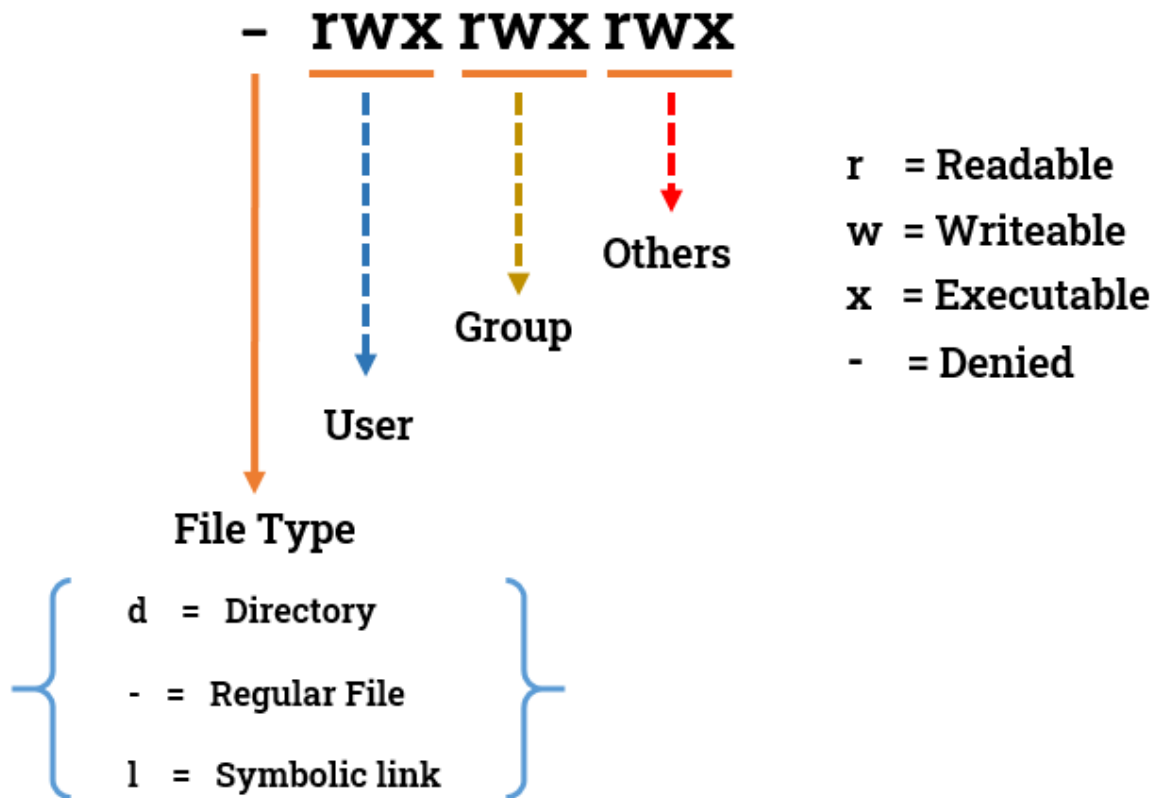
Permet d'afficher le ou les fichiers avec leurs droits associés :

```
ls -l
total 38588
-rw-r--r-- 1 elieroc elieroc 165711 7 oct. 21:32 acl.jpeg
-rw-r--r-- 1 elieroc elieroc 133769 3 oct. 20:44 AGEFIPH.pdf
-rw-r--r-- 1 elieroc elieroc 339479 25 sept. 23:03 'Block devices.pdf'
-rw-r--r-- 1 elieroc elieroc 29345747 27 sept. 20:12 'CCNA Trimestre 1 Partie 1.pdf'
-rw-r--r-- 1 elieroc elieroc 296675 7 oct. 21:32 cron.jpeg
-rw-r--r-- 1 elieroc elieroc 546179 26 sept. 21:50 'DNS Bind.pdf'
-rw-r--r-- 1 elieroc elieroc 1630995 7 oct. 21:31 filesystem.gif
-rw-r--r-- 1 elieroc elieroc 126952 7 oct. 21:39 gnu.png
-rw-r--r-- 1 elieroc elieroc 32858 3 oct. 20:40 'ITRUST - Analyste Threat Intelligence H_F.docx'
-rw-r--r-- 1 elieroc elieroc 1597123 7 oct. 21:31 kerberos.gif
-rw-r--r-- 1 elieroc elieroc 319746 25 sept. 22:20 'Le pain quotidien.pdf'
-rw-r--r-- 1 elieroc elieroc 144785 27 sept. 22:24 marteau.jpg
-rw-r--r-- 1 elieroc elieroc 4428842 25 sept. 21:44 RAID.pdf
-rw-r--r-- 1 elieroc elieroc 259856 25 sept. 21:51 'Système de fichier.pdf'
drwxr-xr-x 1 elieroc elieroc 1054 25 sept. 22:22 'TP remise à niveau + corrigés'
```

```
-rw-r--r-- 1 elieroc elieroc 93176 25 sept. 22:22 'TP remise à niveau + corrigés.zip'
-rw-r--r-- 1 elieroc elieroc 17688 7 oct. 21:40 tux.webp
```

La première colonne indique les droits et le type de fichier, la troisième indique l'utilisateur propriétaire et la quatrième le groupe propriétaire.

Voici le fonctionnement de la première colonne :



Chown

Permet de définir le propriétaire du fichier.

```
chown [-R] <USER>:<GROUP> <FILE|DIRECTORY>
```

Remarque : L'option **-R** permet la récursivité et ainsi d'appliquer l'opération à tous les fichiers et dossiers contenus dans le dossier indiqué.

Chmod

Permet de définir les droits du fichier.

```
chmod [-R] <RIGHTS> <FILE>
```


Le champ RIGHTS doit être remplacé par un motif spécifique désignant les droits que vous souhaitez affecter :

drwxrwxrwx

d = Directory
r = Read
w = Write
x = Execute

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

chmod 777



rwX | rwX | rwX
Owner | Group | Others

Voici un exemple :

```
chmod -R w-- myDirectory/
```

Ici, on autorise seulement le propriétaire à pouvoir modifier tous les fichiers et dossiers contenus dans le dossier **myDirectory** (et le dossier lui-même).

ACL

Ils permettent de définir des droits spécifiques à des utilisateurs ou des groupes d'utilisateurs :

ACL 101

1. ACL is a superset of the regular permissions.

First, create a directory...

```
$ mkdir dir
```

Regular permissions:

```
$ ls -ld dir
drwx-----
```

ACL permissions:

```
$ getfacl dir
USER:    me    rwx
GROUP:   me    ---
OTHER:   ---
```

Both permissions are the same,
let's change that...

2. ACL allows adding permission for specific users and groups

```
$ setfacl -m u:dude:r dir
```

Regular permissions:

```
$ ls -ld dir
drwxr--r--+
```

ACL exists

ACL permissions:

```
$ getfacl dir
USER:    me    rwx
user:    dude  r--
GROUP:   me    ---
mask:    r--
OTHER:   ---
```

This is now the "mask",
aka the union of:

- GROUP,
- any custom ACL user,
- any custom ACL group.

- The "mask" is the mechanism that keeps "regular permissions" and ACL in sync.
- It's updated automatically when running 'chmod' or 'setfacl'

Revision #1

Created 7 October 2023 20:01:33 by Elieroc

Updated 7 October 2023 20:21:24 by Elieroc