les droits sous linux

- Création par : lagrenouille
- Objet : du tuto : les droits des fichiers
- Niveau requis :
  débutant, avisé
- Commentaires : *Contexte d'utilisation du sujet du tuto.* Fix Me!
- Débutant, à savoir : Utiliser GNU/Linux en ligne de commande, tout commence là !.
- * Suivi:
  à-tester

Introduction

Sous un système GNU/Linux tout est fichier, y compris les fichiers spéciaux qui désignent les périphériques.

Linux divise les fichiers en plusieurs catégories :

les répertoires

les fichiers ordinaires (programmes, fichiers de configuration, fichiers de données, etc..)

les fichiers spéciaux (type bloc ou caractère)

tout fichier se voit attribuer des droits pour 3 identités :

Par défaut il n'est possible de permettre ou d'interdire la lecture, l'écriture et l'exécution de fichiers que pour trois catégories d'utilisateurs :

le propriétaire du fichier, le groupe auquel appartient le propriétaire et tous les autres.

le propriétaire - c'est l'utilisateur qui a créé le fichier ou l'utilisateur que root a désigné comme propriétaire

le groupe (qui n'est pas forcément le groupe du propriétaire)

les autres(ceux qui ne font pas partie du groupe)

La commande **ls -lha ou exa -Glia** nous permet d'afficher les droits d'un fichier)

Pour chaque identité (voir plus haut).

il existe 3 droits d'accès :

r	w	x
lecture	écriture	execution

Correspondances des droits en binaire/octale et leurs significations

binaire	octale	droits	explication

000	0	—	aucun droits
001	1	–x	executable
010	2	-w-	ecritureb
011	3	-wx	ecrire + executer
100	4	r–	lire
101	5	r-x	lire + executer
110	6	rw-	lire + ecrire
111	7	rwx	lire-ecrire-executer

Utilisation

Représentés par une chaîne de 9 caractères, regroupés 3 par 3 (rwx rwx rwx), définissent les droits des 3 identités (propriétaire, groupe et les autres).

Parmi les nombreuses options de la commande chmod

-v pour verbose (affichage sur la sortie standard STDOUT du résultat de la commande)

-R traiter les répertoires de façon récursive (application de la commande à l'arborescence entière du répertoire en question)

1. Il y a deux modes d'utilisation de la commande chmod :

de façon littérale

de façon numérique

voir man chmod pour plus d'explication

Exemple sur un répertoire la commande chmod (CHangeMODe) permet de définir et de changer les droits du répertoire et de tous ses sous-répertoires et ses fichiers

777 donne tous les droits au répertoire ainsi que tous les fichiers de ce répertoire, la commande 777 est donc à éviter.

```
Chmod -R 777 tutos/
:~$ ls -la tutos/
drwxrwxrwx  5 momo momo 4096  1 mai   19:57 .
drwxrwxrwx 12 momo momo 4096  1 mai   19:57 blogmiao
drwxrwxrwx 11 momo momo 4096  1 mai   19:59 tercop
```

Interprétation d'une permission 755 ou -rwxr-xr-x :

u : un utilisateur existant dans /etc/passwd

g : un groupe existant valide de /etc/group

o : les autres

*Notation symbolique et octale

Les permissions sont soit la lecture read=r, l'écriture write=w et l'exécution .

read=r	write=w	execute=x
symbole octale	symbole octale	symbole octale
5 binaire de 100	2 binaire de 10	1 binaire de 001
7	5	5
u = utilisateur	g = groupe	o = autres
4+2+1=7	4+1=5	4+1=5

Les systèmes UNIX créent des fichiers et répertoires avec des permissions standard comme suit :

Fichiers | 666 -rw-rw-rw- (6+6+6)

Répertoires | 755 -rwxr-xr-xr (7+5+5) • **644 = rw-r—r—** Lecture, écriture pour le propriétaire / Lecture pour les autres

• **666 = rw-rw-rw-** Lecture, écriture pour tout le monde

• **700 =rwx——** Lecture, écriture, exécution juste pour le propriétaire

• **705 =rwx—r-x** Le propriétaire à tous les droits / Le groupe aucun / Les autres lire et exécuter

• **755 =rwxr-xr-x** Le propriétaire à tous les droits / Les autres lire et exécuter

• **764 =rwxrw-r—** Tous droits pour le propriétaire / Lecture, écriture pour le groupe / Lecture seule pour les autres

• **774 =rwxrwxr—** Tous les droits pour le propriétaire et le groupe / Lecture seule pour les autres

• **775 =rwxrwxr-x** Tous les droits pour le propriétaire et le groupe / Lecture et exécution pour les autres

Pour changer le propriétaire d'un fichier il faut utiliser la commande **chown**

```
chown usager fichier
```

Votre fichier appartient à root, et vous voulez qu'i appartienne à toto

```
chown  toto:toto dossier
```

la commande **chgrp**

Pour changer le groupe d'un fichier :

```
chgrp groupe fichier
```

.

La commande « chgrp » est utilisée pour changer le groupe de fichier ou du répertoire

Seul l'utilisateur de la racine peut changer les attributs/processus du fichier

– R Changer l'autorisation sur tous les sous-répertoires du répertoire(et leurs fichiers)

-c Changer l'autorisation pour chaque fichier

chmod 755=Droits de votre utilisateur : Tous les droits (7)

Droits de votre groupe : Lecture et exécution (5)

Droits de tous les utilisateurs : Lecture et exécution (5)

Une bonne protection, pour que personne vienne écrire sur votre fichier,

même pas vous, c'est un chmod 444

Autres extensions

chmod 666 signifie que tous les utilisateurs peuvent lire et écrire, mais ne s'exécute pas chmod 744 permet seulement au propriétaire de faire toutes les actions, le groupe et les autres ne sont autorisés que pour lire

chmod 711 permet seulement au propriétaire de faire toutes les actions, le groupe et les autres ne sont autorisés que pour lire

chmod 444 Permettre l'autorisation de lecture pour le propriétaire et tous les autres

Les permissions disponibles pour chaque personne / groupe sont les suivantes :

lecture: Donne le droit de lister (nécessite aussi le droit exécution) et lire dans un répertoire, et/ou lire un fichier.

écriture: Donne le droit de créer, modifier, renommer, supprimer des fichiers et/ou répertoires.

exécution: Pour un répertoire : donne le droit de le traverser pour lire ses sous-répertoires.

Pour un fichier : donne le droit de l'exécuter si c'est un programme ou un script

Pour en finir : les droits en chiffres :

• "4"	pour le droit de lecture (read)
• "2"	pour le droit d'écriture (write)
• "1"	pour le droit d'exécution (execute)

allons plus loin

Les droits spécifiques. Cette gestion des droits n'étant pas suffisante pour certains admins, ceux utilisent **les ACL**. une autre gestion des droits. (L access control list ou Listes de contrôle d'accès..

Les listes de contrôle d'accès (ACL) permettent aux administrateurs de logiciels à usages collaboratifs, de donner à certains utilisateurs ou groupes d'utilisateurs le droit d'effectuer certaines actions

(lire, écrire, supprimer) sur des pages déterminées.

on peut donc cacher ou rendre accessibles une ou plusieurs parties d'un logiciels à un groupes ou à un utilisateur, autoriser ou non des accès, des écritures, des suppressions …etc ….

Nous venons de voir les droits avec les extensions de fichiers suivants :

Normalement acl est installé par défaut, ou tout au moins si l'on a installé un serveur.

si je veux savoir les droits sur mon répertoire images:

```
 getfacl  images/
```

```
file: images/
owner: www-data
group: www-data
user::rwx
group::r-x
other::r-x
```

Autres extensions

lrwxrwxrwx	lien symbolique
brw-rw—	periphérique de type blocs
crw-rw-rw-	périphérique de type caractère
srw	socket
prw	pile fifo
Nrwxrw—	certains fichiers réseau

```
ls -lha diaporama
```

lrwxrwxrwx 1 root root 25 4 juin 2020 diaporama → /home/malignum/diaporama/ ls -lha /dev/sda brw-rw—- 1 root disk 8, 0 6 mai 15:16 /dev/sda

getfacl /dev/sda

getfacl : suppression du premier « / » des noms de chemins absolus

file: dev/sda

owner: root

group: disk

user::rw-

group::rw-

other::—

le c pour les périphériques de type caractère (ou nœud de périphérique en mode caractère)

```
ls -lisha /dev/tty
11 0 crw-rw-rw- 1 root tty 5, 0 14 mai   08:19 /dev/tty
```

– le s pour les périphériques de type socket

– le s pour droit SGID ou le droit SUID

Selon le placement du s

rwsrwxrwx, il s'agit d'un droit SUID

rwxrwsrwx, il s'agit d'un droit SGID

Le droit SUID permet d'exécuter un fichier avec les droits du propriétaire du fichier..il faut donc avoir les droits d'exécution, bien souvent ce sera les droits root, donc à utiliser avec précaution…

Le droit SUID est noté —s—— dans le cas ou s remplace un - ou —S——, dans le cas ou s cache un x

Sur un répertoire, ce droit permet d'affecter les droits du ropriétaire à tous les fichiers créés dans ce répertoire.

Le droit SGID permet d'exécuter un fichier avec les droits du groupe propriétaire du fichier.

Le sticky bit (bit collant)

le droit d'écriture signifie que l'on peut créer et supprimer les fichiers de ce répertoire.

Le sticky bit permet de faire la différence entre les deux droits.

Lorsque ce droit est positionné sur un répertoire, il interdit la suppression des fichiers qu'il contient à tout utilisateur autre que le propriétaire.

le Sticky Bit est une autorisation spéciale qui peut être définie sur un répertoire doté d'autorisations « d'écriture »

définies pour le groupe qui y a accès. Ce bit garantit que tous les membres du groupe peuvent écrire dans le répertoire, mais seule la personne qui a créé un fichier, c'est-à-dire le propriétaire du fichier, peut supprimer le fichier.

il est représenté par la lettre t ou T, qui vient remplacer le droit d'exécution x, des autres utilisateurs que le propriétaire et ceux appartenant au groupe du fichier.

———t dans le cas ou t remplace un - ou ———T, dans le cas ou t cache un x.

Ce droit indique que le fichier doit rester en mémoire vive, même si l'on en a plus besoin

Le sticky bit empêche de supprimer des fichiers, il n'empêche pas de les vider de leur contenu

exemple:

je crée le répertoire "rapeteur"

```
mkdir rapeteur
```

```
chmod +t rapeteur
```

```
 ls -lha rapeteur/
total 16K
drwxr-xr-t  2 momo momo 4,0K 16 mai   17:12 .
drwxr-xr-x 53 momo momo  12K 16 mai   17:12
```

seule momo pourra supprimer les fichiers dans rapeteur

```
chmod o-x rapeteur/
```

```
ls -lha rapeteur/
total 16K
drwxr-xr-T  2 momo momo 4,0K 16 mai   17:12 .
drwxr-xr-x 53 momo momo  12K 16 mai   17:12 .
```

la même commande que chmod +t avec :

```
chmod 1755 rapeteur/
```

```
ls -lha rapeteur/
total 16K
drwxr-xr-t  2 momo momo 4,0K 16 mai   17:12 .
drwxr-xr-x 53 momo momo  12K 16 mai   17:12 ..
```

exemple du fichier tmp

```
ls -la / |grep tmp
drwxrwxrwt  16 root root 12288 15 mai   00:20 tmp
```

les 2 commande setfacl (affecter) et getfacl (afficher) qui gèrent les ACL.

Les paramètres les plus utiles sont -s (attribuer), -m (modifier), et -x (supprimer).

u (utilisateur), g (groupe) et o (autres) sont les sujets classiques des opérations des droits d'accès.

Les permissions restent les permissions classiques sous Unix à savoir r, w et x.

Il y a toutes les options suivantes :

-s	–set=aclset the ACL of file(s), replacing the current ACL
-S	–set-file=file read ACL entries to set from file
-m	–modify=acl modify the current ACL(s) of file(s)
-M	–modify-file=file read ACL entries to modify from file
-x	–remove=acl remove entries from the ACL(s) of file(s)
-X	–remove-file=file read ACL entries to remove from file
-b	–remove-all remove all extended ACL entries
-K	–remove-default remove the default ACL
-n	–no-mask don't recalculate the effective rights mask
-d	–default operations apply to the default ACL

-R	–ecursive recurse into subdirectories –post-order visit subdirectories first
-L	–logical logical walk, follow symbolic links
-P	–physical physical walk, do not follow symbolic links restore=file restore ACLs (inverse of `getfacl -R') test test mode (ACLs are not modified)
-v	–version print version and exit
-h	help this help text

Nous avons bien 2 façons de lire les droits sur un fichier

avant toute chose:

```
apt update && apt-get install acl
```

```
ls -lisha /usr/sbin/biosdecode
1182299 24K -rwxr-xr-x 1 root root 24K 17 mai    2021 /usr/sbin/biosdecode
```

```
getfacl /usr/sbin/biosdecode
getfacl : suppression du premier « / » des noms de chemins absolus
# file: usr/sbin/biosdecode
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
```

Nous avons vu comment donner les droits avec la commande “chmod”

Voyons maintenant avec la commande “setfacl” **les commandes setfacl et getfacl**

Exemple: je crée un répersoire de travail collaboratif< /br> des groupe et je donne les droits à chacun

```
sudo mkdir /opt/partage
ls -lisha /opt/partage/
total 8,0K
1331073 4,0K drwxr-xr-x 2 root root 4,0K 15 mai   23:45 .
1308162 4,0K drwxr-xr-x 3 root root 4,0K 15 mai   23:45 ..
```

Visualiser les permissions ACLs du dossier :

```
getfacl /opt/partage/
getfacl : suppression du premier « / » des noms de chemins absolus
# file: opt/partage/
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
```

```
sudo setfacl -m g:momo:rwx /opt/partage/
```

```
touch /opt/partage/firstfichier.txt
```

```
getfacl /opt/partage/firstfichier.txt
getfacl : suppression du premier « / » des noms de chemins absolus
# file: opt/partage/firstfichier.txt
# owner: momo
# group: momo
user::rw-
group::r--
other::r--
```

Suppression juste pour le répertoire

```
setfacl -b /opt/partage/firstfichier
```

Suppression juste pour le fichier

```
setfacl -b fichier  /opt/partage/
```

Pour supprimer les acl par défaut :

setfacl -k repertoire

Suppression pour le répertoire, les sous-répertoires

setfacl -R -k repertoire

Les ACLs par défaut permettent de donner des permissions ACL en héritage pour tout sous-répertoire ou fichier créé dans un répertoire. Toutefois, ces ACLs par défaut ne s’appliquent pas aux objets déjà présents dans le répertoire.

Dans la configuration d’un partage avec des accès multiples, il sera donc nécessaire de procéder en deux étapes :

Modifier l’ACL des fichiers existants, Appliquer un ACL par défaut

Pour définir une ACL par défaut, veuillez ajouter d: avant la règle et spécifiez un répertoire à la place d'un nom de fichier.

```
setfacl -m d:o:rx /répertoire
```

lecture et exécution modifiés pour les utilisateurs du groupe

quelques explications sur kezako: id, inodes, soket, umask

Le système de fichiers Linux est généralement une couche intégrée d’un système d’exploitation Linux utilisée pour gérer la gestion des données du stockage. Cela aide à organiser le fichier sur le stockage sur disque. Il gère le nom du fichier, la taille du fichier, la date de création et bien plus d’informations sur un fichier.

Dans les systèmes de type Linux et Unix, chaque processus se voit attribuer un ID de processus, ou PID. C'est ainsi que le système d'exploitation identifie et garde une trace des processus. … Les processus parents ont un PPID, que vous pouvez voir dans les en-têtes de colonne dans de nombreuses applications de gestion de processus, y compris top , htop et ps .

Un nœud d'index ou inode (contraction de l'anglais index et node) est une structure de données contenant des informations à propos d'un fichier ou répertoire stocké dans certains systèmes de fichiers.

Un inode est un numéro unique qui référence un ou plusieurs fichiers dans le système de fichier. Ce numéro est l'élément de localisation du fichier sur le système de fichiers L'inode est l'endroit où sont stockées toutes les métadonnées d'un fichier : son heure de modification, ses autorisations, etc.

Un socket est un fichier permettant aux processus d'échanger des données.

Un port est une construction logique attribuée aux processus réseau afin qu'ils puissent être identifiés dans le système.

Un socket est une combinaison de port et d'adresse IP. … Le mot « Socket » est la combinaison du port et de l'adresse IP.

Les sockets vous permettent d'échanger des informations entre les processus sur la même machine ou sur un réseau, de répartir

le travail sur la machine la plus efficace et d'accéder facilement aux données centralisées. …

Les interfaces de programme d'application (API) socket sont la norme de réseau pour TCP/IP.

umask Sur les systèmes d'exploitation Linux, tous les nouveaux fichiers sont créés avec un ensemble d'autorisations par défaut.

L'utilitaire umask vous permet d'afficher ou de définir le masque de création en mode fichier, qui détermine les bits d'autorisation pour les fichiers ou répertoires nouvellement créés.

Par défaut, les autorisations de création par défaut sont 666 pour les fichiers

ce qui donne l'autorisation de lecture et d'écriture à l'utilisateur, au groupe et aux d'autres

Lorsqu'un processus crée un nouvel objet de système de fichiers, tel qu'un fichier ou un répertoire,

l'objet se voit attribuer un ensemble d'autorisations par défaut qui est masqué par le umask

Par défaut un répertoire est à 755

Vous pouvez afficher la valeur actuelle du masque utilisateur en en tapant umask puis enter

```
umask
```

0022

umask 022 – Attribue des autorisations pour que vous ayez un accès en lecture/écriture pour les fichiers et en lecture/écriture/recherche pour les répertoires que vous possédez. Tous les autres ont

un accès en lecture uniquement à vos fichiers et un accès en lecture/recherche à vos répertoires.

umask 077. autoriser les autorisations de lecture, d’écriture et d’exécution pour le propriétaire du fichier, mais interdire les autorisations de lecture, d’écriture et d’exécution pour tous les autres

liens

droits-unix https://debian-facile.org/doc:systeme:droits-unix

droits unix-bishttps://debian-facile.org/doc:systeme:droits-unix-bis

chmodhttps://debian-facile.org/doc:systeme:chmod?s[]=chmod

chownhttps://debian-facile.org/doc:systeme:chown