Niveau : Admin Junior	Numéro de la Leçon	Dernière Modification
2/4	<progrecss 3/12 style=inline />	2020/01/30 03:28

Gestion des Droits

Préparation

Dans votre répertoire personnel, créez un fichier tux.jpg grâce à la commande **touch**:

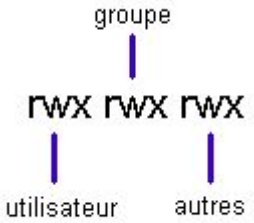
```
$ touch tux.jpg [Entrée]
```

```
trainee@debian:~$ pwd
/home/trainee
trainee@debian:~$ touch tux.jpg
trainee@debian:~$ ls -l | grep tux.jpg
-rw-r--r--. 1 trainee trainee    0 16 oct.  16:24 tux.jpg
```

<note important> Notez que le fichier créé est un fichier **texte**. En effet, Linux ne tient pas compte de l'extension **.jpg** </note>

Les Droits Unix Simples

Les autorisations ou droits d’accès en Linux sont communiqués comme suit :



;#; ;#;

ou r = lecture, w = écriture et x = exécutable

Dans chaque inode est stocké le numéro de l'utilisateur à qui appartient le fichier concerné ainsi que le numéro du groupe. Quand le fichier est ouvert le système compare le numéro de l'utilisateur (UID) avec le numéro de l'utilisateur stocké dans l'inode (Utilisateur de Référence). Si ces deux numéros sont identiques, l'utilisateur obtient les droits du propriétaire du fichier. Si les numéros diffèrent, le système vérifie si l'utilisateur est dans le groupe référencé dans l'inode. Si oui, l'utilisateur aura les droits spécifiés pour le groupe. Si aucune condition n'est remplie, l'utilisateur se voit attribuer les droits des «autres».

Les droits pour les répertoires sont légèrement différents :

r	Les éléments du répertoire sont accessible en lecture (lister)
w	Les éléments du répertoire sont modifiables (création et suppression).
x	Le nom du répertoire peut apparaître dans un chemin d'accès.

La Modification des Droits

La Commande chmod

Les options de cette commande sont :

```
trainee@debian:~$ chmod --help
Utilisation : chmod [OPTION]... MODE[,MODE]... FILE...
         ou : chmod [OPTION]... OCTAL-MODE FILE
         ou : chmod [OPTION]... --reference=RFILE FILE
Change le mode de chaque FILE en MODE.
```

```
  -c, --changes           comme « verbose » mais affiche seulements les
                          changements réalisés
      --no-preserve-root  ne traite pas « / » de manière spéciale (par défaut)
      --preserve-root     bloque le traitement récursif sur « / »
  -f, --silent, --quiet   supprime la plupart des messages d'erreur
  -v, --verbose           produit un diagnostic pour chaque fichier traité
      --reference=RFILE   utilise le mode RFILE au lieu des valeurs MODE
  -R, --recursive         modifie récursivement les fichiers et les répertoires
      --help     affiche l'aide et quitte
      --version  affiche des informations de version et quitte

Chaque MODE est de la forme « [ugoa]*([-+=]([rwxXst]*|[ugo]))+ ».

Signalez les anomalies de « chmod » à <bug-coreutils@gnu.org>
Page d'accueil de « GNU coreutils » : <http://www.gnu.org/software/coreutils/>
Aide générale sur les logiciels GNU : <http://www.gnu.org/gethelp/>
Traduction de « chmod » à <http://translationproject.org/team/fr.html>
Pour une documentation complète, lancer « info coreutils 'chmod invocation' »
```

Mode Symbolique

Afin de modifier les droits d’accès aux fichiers, on utilise la commande chmod dont le syntaxe est le suivant :

chmod [-R] catégorie opérateur permissions nom_du_fichier

ou

chmod [-R] ugoa +-= rwxXst nom_du_fichier

où

u	user
g	group

o	other
a	all
+	autorise un accès
-	interdit un accès
=	autorise exclusivement l'accès indiqué
r	read
w	write
x	execute
X	exécution si la cible est un répertoire ou si c'est un fichier est déjà exécutable pour une des *catégories* (ugo)
s	SUID/SGID bit
t	sticky bit

par exemple :

```
$ chmod o+w tux.jpg [Entrée]
```

donnera aux autres l'accès en écriture sur le fichier tux.jpg :

```
trainee@debian:~$ chmod o+w tux.jpg
trainee@debian:~$ ls -l | grep tux.jpg
-rw-r--rw-. 1 trainee trainee    0 16 oct.  16:24 tux.jpg
```

Tandis que :

```
$ chmod ug-w tux.jpg [Entrée]
```

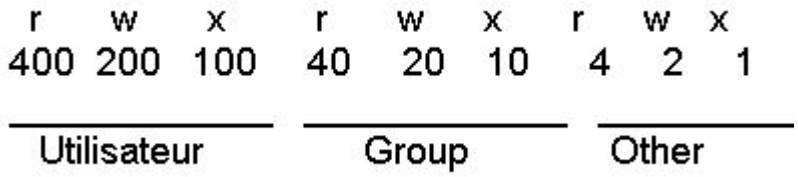
ôtera les droit d'accès en écriture pour l'utilisateur et le groupe :

```
trainee@debian:~$ chmod ug-w tux.jpg
trainee@debian:~$ ls -l | grep tux.jpg
-r--r--rw-. 1 trainee trainee    0 16 oct.  16:24 tux.jpg
```

<note tip> Seul le propriétaire du fichier ou root peuvent modifier les permissions. </note>

Mode Octal

La commande chmod peut également être utilisée avec une représentation octale (base de 8). Les valeurs octales des droits d'accès sont :



;#; ;#;

<note important> Ainsi les droits rwx rwx rwx correspondent à un chiffre de 777. </note>

La commande chmod prend donc la forme suivante:

```
chmod [ -R ] mode_octal nom_fichier
```

La commande suivante :

```
$ chmod 644 tux.jpg [Entrée]
```

Correspond donc à l'attribution des droits : rw- r– r–

```
trainee@debian:~$ chmod 644 tux.jpg
trainee@debian:~$ ls -l | grep tux.jpg
-rw-r--r--. 1 trainee trainee    0 16 oct.  16:24 tux.jpg
```

<note important> Les droits d'accès par défaut lors de la création d'un objet sont :

Répertoires	rwx rwx rwx	777

Fichier normal	rw– rw– rw-	666

</note>

La Commande umask

L’utilisateur peut changer ces droits d’accès par défaut lors de la création d’objets en utilisant la commande umask. Les options de la commande sont détaillées ci-après :

```
trainee@debian:~$ help umask
umask: umask [-p] [-S] [mode]
    Display or set file mode mask.
    Sets the user file-creation mask to MODE.  If MODE is omitted, prints
    the current value of the mask.
    If MODE begins with a digit, it is interpreted as an octal number;
    otherwise it is a symbolic mode string like that accepted by chmod(1).
    Options:
      -p    if MODE is omitted, output in a form that may be reused as input
      -S    makes the output symbolic; otherwise an octal number is output
    Exit Status:
    Returns success unless MODE is invalid or an invalid option is given.
```

La valeur par défaut de l'umask sous Debian est indentique pour un utilisateur normal et pour root :

```
trainee@debian:~$ umask
0022
trainee@debian:~$ su -
Mot de passe : fenestros
root@debian:~# umask
0022
root@debian:~# exit
logout
```

Par exemple dans le cas où l'utilisateur souhaite que les fichiers créés dans le futur comportent des droits d'écriture et de lecture pour l'utilisateur mais uniquement des droits de lecture pour le groupe et pour les autres, il utiliserait la commande :

```
$ umask 022 [Entrée]
```

avant de créer son fichier.

<note tip> umask sert à enlever des droits des droits maximaux :

Masque maximum lors de la création d'un fichier	rw- rw- rw-	666
Droits à retirer	— -w- -w-	022
Résultat	rw- r– r–	644

</note>

Dans l'exemple qui suit, on utilise la commande touch pour créer un fichier vide ayant les nouveaux droits par défaut :

```
trainee@debian:~$ umask 044
trainee@debian:~$ touch tux1.jpg
trainee@debian:~$ ls -l | grep tux1.jpg
-rw--w--w-. 1 trainee trainee    0 16 oct.  16:29 tux1.jpg
trainee@debian:~$ umask 022
trainee@debian:~$ umask
0022
```

Modifier le propriétaire ou le groupe

Le changement de propriétaire d'un fichier se fait uniquement par l'administrateur système - root.

La Commande chown

Les options de cette commande sont détaillées ci-après :

```
trainee@debian:~$ chown --help
Utilisation : chown [OPTION]... [OWNER][:GROUP] FILE...
         ou : chown [OPTION]... --reference=RFILE FILE...
Change le propriétaire et/ou le groupe de chaque FILE à OWNER et/ou à GROUP.
Avec --reference, change le propriétaire et le groupe de chaque FILE à ceux de
RFILE.

  -c, --changes           comme verbeux mais rapporte seulement les
                          modifications réalisées
      --dereference       affecte le référent de chaque lien symbolique (par
                          défaut), plutôt que le lien symbolique lui-même
  -h, --no-dereference   affecte les liens symboliques au lieu des fichiers
                         référencés (utile seulement sur les systèmes permettant
                         de changer le propriétaire d'un lien symbolique)
      --from=CURRENT_OWNER:CURRENT_GROUP
                         change the owner and/or group of each file only if
                         its current owner and/or group match those specified
                         here.  Either may be omitted, in which case a match
                         is not required for the omitted attribute
      --no-preserve-root  ne traite pas « / » de manière spéciale (par défaut)
      --preserve-root     bloque le traitement récursif sur « / »
  -f, --silent, --quiet  supprime la plupart des messages d'erreur
      --reference=RFILE  utilise le propriétaire et le groupe RFILE au lieu de
                         valeurs explicites OWNER:GROUP
  -R, --recursive        agit récursivement sur les fichiers et les répertoires
  -v, --verbose          affiche un diagnostique pour chaque fichier traité

Les options suivantes modifient la façon dont la hiérarchie est traversée
lorsque l'option -R est aussi spécifiée. Si plusieurs options sont indiquées,
```

```
seule la dernière sera prise en compte.

  -H                     si l'argument en ligne de commande est un lien
                         symbolique vers un répertoire alors le parcourir
  -L                     parcourt tous les liens symboliques menant à un
                         répertoire
  -P                     ne parcourt aucun lien symbolique (par défaut)

      --help     affiche l'aide et quitte
      --version  affiche des informations de version et quitte

Le propriétaire n'est pas modifié si manquant. Le groupe n'est pas modifié si
manquant, mais changé en groupe de connexion si un « : » suit un symbolique
OWNER (propriétaire).
Le OWNER et le GROUP peuvent être numériques ou symboliques.

Exemples :
  chown root /u        change le propriétaire de /u en « root ».
  chown root:staff /u  idem mais change aussi son groupe en « staff ».
  chown -hR root /u    change le propriétaire de /u et des sous-fichiers en
                   « root ».

Signalez les anomalies de « chown » à <bug-coreutils@gnu.org>
Page d'accueil de « GNU coreutils » : <http://www.gnu.org/software/coreutils/>
Aide générale sur les logiciels GNU : <http://www.gnu.org/gethelp/>
Traduction de « chown » à <http://translationproject.org/team/fr.html>
Pour une documentation complète, lancer « info coreutils 'chown invocation' »
```

Dans le cas du fichier tux.jpg appartenant à trainee, root peut changer le propriétaire de trainee à root avec la commande suivante :

```
# chown root tux.jpg [Entrée]
```

```
trainee@debian:~$ su -
Mot de passe :
```

```
root@debian:~# cd /home/trainee
root@debian:/home/trainee# chown root tux.jpg
root@debian:/home/trainee# ls -l | grep tux.jpg
-rw-r--r--. 1 root    trainee    0 16 oct.  16:24 tux.jpg
```

La Commande chrgp

Les options de cette commande sont détaillées ci-après :

```
root@debian:/home/trainee# chgrp --help
Utilisation : chgrp [OPTION]... GROUP FILE...
         ou : chgrp [OPTION]... --reference=RFILE FILE...
Change le groupe de chaque FILE en GROUP.
Avec l'option --reference, change le groupe de chaque FILE à celui de RFILE.

  -c, --changes          comme l'option « verbose » mais affiche seulement les
                         changements effectués
      --dereference      affecte le référent de chaque lien symbolique (par
                         défaut), plutôt que le lien symbolique lui-même
  -h, --no-dereference   affecte les liens symboliques au lieu des fichiers
                         référencés (utile seulement sur les systèmes permettant
                         de changer le propriétaire d'un lien symbolique)
      --no-preserve-root  ne traite pas « / » de manière spéciale (par défaut)
      --preserve-root     bloque le traitement récursif sur « / »
  -f, --silent, --quiet  supprime la plupart des messages d'erreur
      --reference=RFILE  utilise le groupe RFILE au lieu d'une valeur GROUPE
  -R, --recursive        agit récursivement sur les fichiers et répertoires
  -v, --verbose          produit un diagnostic pour chaque fichier traité

Les options suivantes modifient la façon dont la hiérarchie est traversée
lorsque l'option -R est aussi spécifiée. Si plusieurs options sont indiquées,
seule la dernière sera prise en compte.
```

```
  -H                     si l'argument en ligne de commande est un lien
                         symbolique vers un répertoire alors le parcourir
  -L                     parcourt tous les liens symboliques menant à un
                         répertoire
  -P                     ne parcourt aucun lien symbolique (par défaut)

      --help     affiche l'aide et quitte
      --version  affiche des informations de version et quitte

Exemples :
  chgrp staff /u      change le groupe de /u en « staff ».
  chgrp -hR staff /u  change le groupe de /u et des sous-fichiers en « staff ».

Signalez les anomalies de « chgrp » à <bug-coreutils@gnu.org>
Page d'accueil de « GNU coreutils » : <http://www.gnu.org/software/coreutils/>
Aide générale sur les logiciels GNU : <http://www.gnu.org/gethelp/>
Traduction de « chgrp » à <http://translationproject.org/team/fr.html>
Pour une documentation complète, lancer « info coreutils 'chgrp invocation' »
```

Le même cas de figure s'applique au groupe :

```
# chgrp root tux.jpg [Entrée]
```

affectera le fichier au groupe root :

```
root@debian:/home/trainee# chgrp root tux.jpg
root@debian:/home/trainee# ls -l | grep tux.jpg
-rw-r--r--. 1 root    root       0 16 oct.  16:24 tux.jpg
```

<note important> Seul root peut changer le propriétaire d'un fichier. </note>

<note important> Le droit de supprimer un fichier dépend des droits sur le répertoire dans lequel le fichier est stocké et non des droits du fichier lui-même. </note>

Les Droits Unix Etendus

SUID/SGID bit

Malgré ce que vous venez de voir, dans la première des deux fenêtres ci-dessous, vous noterez que le fichier **passwd** se trouvant dans le répertoire **/etc** possède les permissions **rw- r– r–** et qu'il appartient à **root**. Autrement dit **seul** root peut écrire dans ce fichier. Or, quand un utilisateur normal change son mot de passe, il écrit dans ce fichier. Ceci semble donc être une contradiction.

```
root@debian:/home/trainee# ls -l /etc/passwd /usr/bin/passwd
-rw-r--r--. 1 root root  1298 27 avril 16:36 /etc/passwd
-rwsr-xr-x. 1 root root 34740 15 févr.  2011 /usr/bin/passwd
```

Pour remédier à cette apparente contradiction, Linux dispose de deux droits d'accès étendus :

- Set UserID bit (SUID bit)
- Set GroupID bit (SGID bit)

Quand le SUID bit est placé sur un programme, l'utilisateur qui lance ce programme se voit affecté le numéro d'utilisateur du propriétaire de ce programme et ce pour la durée de son exécution.

Dans le cas du changement de mot de passe, chaque utilisateur qui lance le programme /usr/bin/passwd se trouve temporairement avec le numéro d'utilisateur du propriétaire du programme /usr/bin/passwd, c'est à dire root. De cette façon, l'utilisateur peut intervenir sur le fichier /etc/passwd. Ce droit est indiqué par la lettre s à la place de la lettre x.

La même fonction existe pour le groupe à l'aide du SGID bit.

Pour assigner les droits, vous utiliserez la commande chmod :

- chmod u+s nom_du_fichier
- chmod g+s nom_du_fichier

En base huit les valeurs sont les suivants :

- SUID = 4000
- SGID = 2000

Inheritance Flag

Le SGID bit peut également être affecté à un répertoire. De cette façon, les fichiers et répertoires créés à l'intérieur auront comme groupe le groupe du répertoire parent. Ce droit s'appelle donc l'**Inheritance Flag** ou le **Drapeau d'Héritage**.

Par exemple :

```
root@debian:/home/trainee# cd /tmp
root@debian:/tmp# mkdir inherit
root@debian:/tmp# chown root:trainee inherit
root@debian:/tmp# chmod g+s inherit
root@debian:/tmp# touch inherit/test.txt
root@debian:/tmp# mkdir inherit/testrep
root@debian:/tmp# cd inherit; ls -l
total 4
drwxr-sr-x. 2 root trainee 4096 16 oct.  16:50 testrep
-rw-r--r--. 1 root trainee    0 16 oct.  16:50 test.txt
```

Sticky bit

Il existe un dernier cas qui s’appelle le sticky bit. Le sticky bit est utilisé pour des répertoires ou tout le monde a tous les droits. Dans ce cas, tout le monde peut supprimer des fichiers dans le répertoire. En ajoutant le sticky bit, uniquement le propriétaire du fichier peut le supprimer.

```
# chmod o+t /répertoire
```

ou

```
# chmod 1777 /répertoire
```

Par exemple la ligne de commande:

```
# mkdir /tmp/repertoire_public; cd /tmp; chmod o+t repertoire_public [Entrée]
```

ou

```
# mkdir /tmp/repertoire_public; cd /tmp; chmod 1777 repertoire_public [Entrée]
```

créera un répertoire repertoire_public dans /tmp avec les droits suivants :

```
root@debian:/tmp/inherit# mkdir /tmp/repertoire_public; cd /tmp; chmod o+t repertoire_public
root@debian:/tmp# ls -l | grep repertoire_public
drwxr-xr-t. 2 root       root       4096 16 oct.  16:53 repertoire_public
```

Les Droits Unix Avancés

Les ACL

Au delà des droits étendus d'Unix, Linux utilise un système d'ACL pour permettre une meilleure gestion des droits sur des fichiers.

Sous Debian, il faut installer le paquet **acl** pour pouvoir les utiliser :

```
root@debian:/tmp# apt-get install acl
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les NOUVEAUX paquets suivants seront installés :
  acl
0 mis à jour, 1 nouvellement installés, 0 à enlever et 4 non mis à jour.
Il est nécessaire de prendre 61,7 ko dans les archives.
Après cette opération, 295 ko d'espace disque supplémentaires seront utilisés.
```

```
Réception de : 1 http://ftp.fr.debian.org/debian/ squeeze/main acl i386 2.2.49-4 [61,7 kB]
61,7 ko réceptionnés en 0s (235 ko/s)
Sélection du paquet acl précédemment désélectionné.
(Lecture de la base de données... 144260 fichiers et répertoires déjà installés.)
Dépaquetage de acl (à partir de .../archives/acl_2.2.49-4_i386.deb) ...
Traitement des actions différées (« triggers ») pour « man-db »...
Paramétrage de acl (2.2.49-4) ...
```

Chaque partition sur laquelle vous voulez utiliser les ACLs doit être montée avec l'option ACL. Modifiez donc le fichier **/etc/fstab** ainsi :

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    defaults        0       0
# / was on /dev/sda1 during installation
UUID=a42a1ddd-14bc-4dde-a537-e6c1b984a782 /               ext3    errors=remount-ro,acl 0       1
# swap was on /dev/sda5 during installation
UUID=e21d8931-21ca-4ab3-9fbb-bd71657b312e none            swap    sw              0       0
/dev/scd0       /media/cdrom0   udf,iso9660 user,noauto     0       0
```

Remontez votre système de fichiers racine et vérifiez que l'option **acl** existe :

```
root@debian:/tmp# mount -o remount /
root@debian:/tmp# mount | grep acl
/dev/sda1 on / type ext3 (rw,errors=remount-ro,acl)
```

Pour connaître les ACL positionnés sur un fichier, il convient d'utiliser la commande **getfacl** :

```
# getfacl /home/trainee/tux.jpg [Entrée]
```

Les options de la commande **getfacl** sont :

```
root@debian:/tmp# getfacl --help
getfacl 2.2.49 -- obtenir les listes de contrôle d'accès du fichier
Utilisation : getfacl [-aceEsRLPtpndvh] fichier...
  -a,  --access           display the file access control list only
  -d, --default           display the default access control list only
  -c, --omit-header       do not display the comment header
  -e, --all-effective     print all effective rights
  -E, --no-effective      print no effective rights
  -s, --skip-base         skip files that only have the base entries
  -R, --recursive         recurse into subdirectories
  -L, --logical           logical walk, follow symbolic links
  -P, --physical          physical walk, do not follow symbolic links
  -t, --tabular           use tabular output format
  -n, --numeric           print numeric user/group identifiers
  -p, --absolute-names    don't strip leading '/' in pathnames
  -v, --version           print version and exit
  -h, --help              this help text
```

En utilisant cette commande, vous obtiendrez un résultat similaire à celui-ci :

```
root@debian:/tmp# getfacl /home/trainee/tux.jpg
getfacl : suppression du premier « / » des noms de chemins absolus
# file: home/trainee/tux.jpg
# owner: root
# group: root
user::rw-
group::r--
other::r--
```

Pour positionner des ACL sur un fichier, il convient d'utiliser la commande **setfacl** :

```
# setfacl --set u::rwx,g::rx,o::-,u:trainee:rw /home/trainee/tux.jpg [Entrée]
```

Les options de la commande **setfacl** sont :

```
root@debian:/tmp# setfacl --help
setfacl 2.2.49 -- définir les listes de contrôle d'accès des fichiers (ACL)
Utilisation : setfacl [-bkndRLP] { -m|-M|-x|-X ... } file ...
  -m, --modify=acl           modifier l'ACL(s) actuel de fichier(s)
  -M, --modify-file=fichier  lire l'entrée ACL à modifier du fichier
  -x, --remove=acl           supprimer les entrées de l'ACL des fichier
  -X, --remove-file=fichier  lire les entrées ACL à supprimer du fichier
  -b, --remove-all           supprimer toutes les entrées ACL étendues
  -k, --remove-default       supprimer l'ACL par défaut
      --set=acl           set the ACL of file(s), replacing the current ACL
      --set-file=file     read ACL entries to set from file
      --mask              do recalculate the effective rights mask
  -n, --no-mask           ne pas recalculer les masques de droits en vigueur
  -d, --default           les opérations s'appliquent à l'ACL par défaut
  -R, --recursive         parcourir récursivement les sous-répertoires
  -L, --logical           suivre les liens symboliques
  -P, --physical          ne pas suivre les liens symboliques
      --restore=fichier   restaurer les ACL (inverse de « getfacl -R »)
      --test              mode test (les ACL ne sont pas modifiés)
  -v, --version           print version and exit
  -h, --help              this help text
```

Utilisez la commande **getfacl** pour visualiser le résultat :

```
# getfacl /home/trainee/tux.jpg [Entrée]
```

Vous obtiendrez un résultat similaire à celui-ci :

```
root@debian:/tmp# setfacl --set u::rwx,g::rx,o::-,u:trainee:rw /home/trainee/tux.jpg
```

```
root@debian:/tmp# getfacl /home/trainee/tux.jpg
getfacl : suppression du premier « / » des noms de chemins absolus
# file: home/trainee/tux.jpg
# owner: root
# group: root
user::rwx
user:trainee:rw-
group::r-x
mask::rwx
other::---
```

En effet, tous les utilisateurs ont les permissions **rwx** sauf **trainee**.

Regardez maintenant l'effet des ACL sur un répertoire. Créez le répertoire /home/trainee/rep1 :

```
# mkdir /home/trainee/rep1 [Entrée]
```

Positionnez des ACL le répertoire avec la commande **setfacl** :

```
# setfacl --set d:u::r,d:g::-,d:o::- /home/trainee/rep1 [Entrée]
```

Notez l'utilisation de la lettre **d** pour indiquer une permission *par défaut*.

Créez maintenant un fichier appelé fichier1 dans /home/trainee/rep1 :

```
# touch /home/trainee/rep1/fichier1 [Entrée]
```

Utilisez la commande **getfacl** pour visualiser le résultat :

```
# getfacl /home/trainee/rep1 [Entrée]
```

```
# getfacl home/trainee/rep1/fichier1 [Entrée]
```

Vous obtiendrez un résultat similaire à celui-ci :

```
root@debian:/tmp# mkdir /home/trainee/rep1
root@debian:/tmp# setfacl --set d:u::r,d:g::-,d:o::- /home/trainee/rep1
root@debian:/tmp# touch /home/trainee/rep1/fichier1
root@debian:/tmp# getfacl /home/trainee/rep1
getfacl : suppression du premier « / » des noms de chemins absolus
# file: home/trainee/rep1
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
default:user::r--
default:group::---
default:other::---

root@debian:/tmp# getfacl /home/trainee/rep1/fichier1
getfacl : suppression du premier « / » des noms de chemins absolus
# file: home/trainee/rep1/fichier1
# owner: root
# group: root
user::r--
group::---
other::---
```

Notez que le fichier créé possède les ACL positionnés sur le répertoire rep1.

Dernièrement, les systèmes de sauvegarde classiques sous Linux ne peuvent pas sauvegarder les ACL, sauf l'outil **star**. Si vous n'utilisez pas **star**, il convient donc de sauvegarder les ACL dans un fichier grâce à la commande suivante :

```
# getfacl -R --skip-base . > backup.acl [Entrée]
```

La restauration des ACL se fait avec la commande **setfacl** :

```
# setfacl --restore=backup.acl [Entrée]
```

<note warning>

```
         mask    A mask ACL entry specifies the maximum access which can be
                 granted by any ACL entry except the user entry for the file owner
                 and the other entry (entry tag type ACL_MASK).
```

</note>

Les Attributs Ext2/Ext3/Ext4

Les attributs s'ajoutent aux caractéristiques classiques d'un fichier dasn un système de fichiers Ext2/Ext3 et ReiserFS.

Les principaux attributs sont :

Attribut	**Description**
a	Fichier journal - uniquement l'ajout de données au fichier est permis. Le fichier ne peut pas détruit
i	Le fichier ne peut ni être modifié, ni être détruit, ni être déplacé. Le placement d'un lien sur le fichier n'est pas permis
s	Le fichier sera physiquement détruit lors de sa suppression
D	Répertoire synchrone
S	Fichier synchrone
A	La date et l'heure de dernier accès ne seront pas mises à jour

<note tip> Un fichier synchrone et un répertoire synchrone impliquent que les modifications seront immédiatement inscrites sur disque. </note>

Les commandes associées avec les attributs sont :

Commande	**description**
chattr	Modifie les attributs
lsattr	Visualise les attributs

Les options de la commande **chattr** sont :

```
root@debian:/tmp# chattr --help
Usage : chattr [-RVf] [-+=AacDdeijsSu] [-v version] fichiers...
```

Les options de la commande **lsattr** sont :

```
root@debian:/tmp# lsattr --help
...
Usage : lsattr [-RVadlv] [fichiers...]
```

Pour mieux comprendre, créez le répertoire **/tmp/attributs/rep** :

```
root@debian:/tmp# mkdir -p attributs/rep
```

Créez ensuite les fichier **fichier** et **rep/fichier1** :

```
root@debian:/tmp# touch attributs/fichier
root@debian:/tmp# touch attributs/rep fichier1
```

Modifiez les attributs d'une manière récursive sur le répertoire **attributs** :

```
root@debian:/tmp# chattr +i -R attributs/
```

Visualisez les attributs de l'arborescence **attributs** d'une manière récursive :

```
root@debian:/tmp# lsattr -R attributs
----i-------------- attributs/fichier
----i-------------- attributs/rep

attributs/rep:
```

Essayez maintenant de déplacer le fichier **fichier**. Vous obtiendrez un résultat similaire à celui-ci :

```
root@debian:/tmp# cd attributs; mv /tmp/attributs/fichier /tmp/attributs/rep/fichier
mv: impossible de déplacer « /tmp/attributs/fichier » vers « /tmp/attributs/rep/fichier »: Permission non
accordée
```

~~DISCUSSION:off~~

Donner votre Avis

{(rater>id=debian_6_l108|name=cette page|type=rate|trace=user|tracedetails=1)}