Dernière mise-à-jour : 2020/08/06 15:18

LDF203 - Gestion des Droits

Contenu du Module

Présentation

Dans sa conception de base, Linux utilise une approche sécurité de type **DAC** (**D**iscretional **A**ccess **C**ontrol). Cette approche est maintenue dans la

mise en place et l'utilisation des **ACL** et les **Attributs Etendus Ext2/Ext3/Ext4, JFS, ReiserFS, XFS et Btrfs** :

Type de Sécurité	Nom	Description
DAC	*Discretional Access Control*	L'accès aux objets est en fonction de l'identité (utilisateur,groupe). Un utilisateur peut rendre accessible aux autres ses propres objets.

Préparation

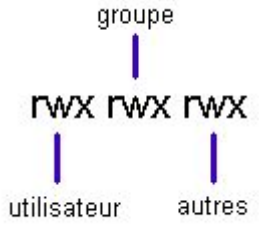
Dans votre répertoire personnel, créez un fichier tux.jpg grâce à la commande **touch**:

```
root@debian9:~# exit
déconnexion
trainee@debian9:~$ pwd
/home/trainee
trainee@debian9:~$ touch tux.jpg
trainee@debian9:~$ ls -l | grep tux.jpg
-rw-r--r-- 1 trainee trainee    0 avril  4 13:54 tux.jpg
```

Important : Notez que le fichier créé est un fichier **texte**. En effet, Linux ne tient pas compte de l'extension **.jpg**

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

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@debian9:~$ chmod o+w tux.jpg
trainee@debian9:~$ ls -l | grep tux.jpg
-rw-r--rw- 1 trainee trainee    0 avril  4 13:54 tux.jpg
```

Tandis que :

```
$ chmod ug-w tux.jpg [Entrée]
```

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

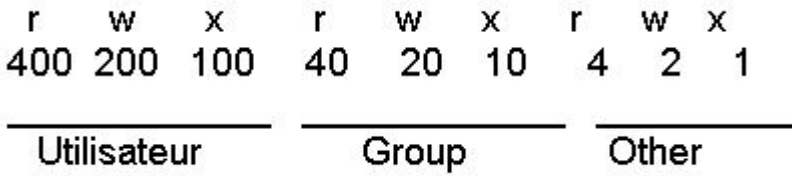
```
trainee@debian9:~$ chmod ug-w tux.jpg
trainee@debian9:~$ ls -l | grep tux.jpg
-r--r--rw- 1 trainee trainee    0 avril  4 13:54 tux.jpg
```

Important : Seul le propriétaire du fichier ou root peuvent modifier les permissions.

Important : Notez que dans le cas du dernier exemple ni le propriétaire (trainee), ni le groupe associé au fichier (trainee ou users) ne peut écrire dans le fichier et ceci malgré le fait que le reste du monde peut y écrire !

Mode Octal

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



Important : Ainsi les droits rwx rwx rwx correspondent à un chiffre de 777.

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@debian9:~$ chmod 644 tux.jpg
trainee@debian9:~$ ls -l | grep tux.jpg
-rw-r--r-- 1 trainee trainee    0 avril  4 13:54 tux.jpg
```

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

Options de la Commande chmod

Les options de cette commande sont :

```
trainee@debian9:~$ chmod --help
Utilisation : chmod [OPTION]... MODE[,MODE]... FICHIER...
         ou : chmod [OPTION]... MODE_OCTAL FICHIER
         ou : chmod [OPTION]... --reference=FICHIER_R FICHIER
Modifier le mode de chaque FICHIER en MODE.
Avec --reference, modifier le mode de chaque FICHIER à celui de FICHIER_R.
  -c, --changes          comme --verbose, mais seulement en cas de modification
  -f, --silent, --quiet  supprimer la plupart des messages d'erreur
  -v, --verbose          afficher un diagnostic pour chaque fichier traité
      --no-preserve-root ne pas traiter « / » de manière spéciale (par défaut)
      --preserve-root    bloquer le traitement récursif sur « / »
      --reference=FICHIER_R  utiliser le mode de FICHIER_R au lieu des valeur
```

```
                              de MODE
  -R, --recursive        modifier récursivement les fichiers et répertoires
      --help     afficher l'aide et quitter
      --version  afficher des informations de version et quitter

Chaque MODE est de la forme « [ugoa]*([-+=]([rwxXst]*|[ugo]))+|[-+=][0-7]+ ».

Aide en ligne de GNU coreutils : <http://www.gnu.org/software/coreutils/>
Signalez les problèmes de traduction de « chmod » à : <traduc@traduc.org>
Full documentation at: <http://www.gnu.org/software/coreutils/chmod>
or available locally via: info '(coreutils) chmod invocation'
```

La Commande umask

L'utilisateur peut changer sa masque de permissions défaut lors de la création d'objets en utilisant la commande umask.

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

```
trainee@debian9:~$ umask
0022
trainee@debian9:~$ su -
Mot de passe : fenestros
root@debian9:~# umask
0022
root@debian9:~# exit
déconnexion
```

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.

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

Modifier la Valeur d'umask

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

```
trainee@debian9:~$ umask 044
trainee@debian9:~$ touch tux1.jpg
trainee@debian9:~$ ls -l | grep tux1.jpg
-rw--w--w- 1 trainee trainee    0 avril  4 13:58 tux1.jpg
trainee@debian9:~$ umask 022
trainee@debian9:~$ umask
0022
```

Options de la Commande

Les options de cette commande sont :

```
trainee@debian9:~$ help umask
umask: umask [-p] [-S] [mode]
    Affiche ou définit le masque de mode de fichier.
    Définit le masque de création de fichier comme étant MODE.  Si MODE est omis,
    affiche la valeur courante du MASQUE.
    Si MODE commence par un chiffre, il est interprété comme un nombre octal ;
    sinon comme une chaîne de symboles de mode comme ceux acceptés par chmod(1).
    Options :
      -p    si MODE est omis, affiche sous une forme réutilisable en entrée
      -S    affiche sous forme symbolique, sinon la sortie octale est utilisée
```

```
    Code de retour :
    Renvoie le code de succès à moins que MODE ne soit pas valable ou qu'une
    option non valable ne soit donnée.
```

Modifier le propriétaire ou le groupe

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

La Commande chown

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@debian9:~$ su -
Mot de passe : fenestros
root@debian9:~# cd /home/trainee
root@debian9:/home/trainee# chown root tux.jpg
root@debian9:/home/trainee# ls -l | grep tux.jpg
-rw-r--r-- 1 root    trainee    0 avril  4 13:54 tux.jpg
```

Options de la Commande

Les options de cette commande sont :

```
root@debian9:/home/trainee# chown --help
Utilisation : chown [OPTION]... [PROPRIO][:GROUPE] FICHIER...
```

```
        ou : chown [OPTION]... --reference=FICHIER_R FICHIER...
Modifier le propriétaire ou le groupe de chaque FICHIER en PROPRIO ou GROUPE.
Avec --reference, modifier le propriétaire et le groupe de chaque FICHIER à
ceux de FICHIER_R.

  -c, --changes          comme --verbose, mais seulement en cas de modification
  -f, --silent, --quiet  supprimer la plupart des messages d'erreur
  -v, --verbose          afficher un diagnostic pour chaque fichier traité
      --dereference      affecter le référent de chaque lien symbolique (par
                           défaut), au lieu du lien symbolique lui-même
  -h, --no-dereference   affecter les liens symboliques au lieu des fichiers
                           référencés
                           (seulement utile sur les systèmes permettant de
                           modifier le propriétaire d'un lien symbolique)
      --from=PROPRIO_ACTUEL:GROUPE_ACTUEL
                         modifier le propriétaire ou le groupe de chaque fichier
                         dont le propriétaire ou le groupe actuel correspondent
                         à ceux indiqués. La correspondance n'est nécessaire que
                         pour l'argument indiqué si l'autre est omis.
      --no-preserve-root ne pas traiter « / » de manière spéciale (par défaut)
      --preserve-root    bloquer le traitement récursif sur « / »
      --reference=FICHIER_R  utiliser les propriétaires et groupe de FICHIER_R
                             au lieu d'indiquer des valeurs PROPRIO:GROUPE
  -R, --recursive        opérer récursivement sur les fichiers et répertoires

Les options suivantes modifient la façon de parcourir la hiérarchie lorsque
l'option -R est aussi indiquée. 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, le parcourir
  -L                     parcourir tous les liens symboliques menant à un
                           répertoire
  -P                     ne parcourir aucun lien symbolique (par défaut)
```

```
      --help     afficher l'aide et quitter
      --version  afficher des informations de version et quitter

Le propriétaire n'est pas modifié s'il n'est pas indiqué. Le groupe n'est pas
modifié sil n'est pas indiqué, mais modifié en groupe de connexion s'il est
sous-entendu par un « : » suivant un PROPRIO symbolique.
Les PROPRIO et GROUPE peuvent être numériques ou symboliques.

Exemples :
  chown root /u        Modifier le propriétaire de /u en « root ».
  chown root:staff /u  Idem mais modifier aussi son groupe en « staff ».
  chown -hR root /u    Modifier le propriétaire de /u et ses sous-fichiers
                         en « root ».

Aide en ligne de GNU coreutils : <http://www.gnu.org/software/coreutils/>
Signalez les problèmes de traduction de « chown » à : <traduc@traduc.org>
Full documentation at: <http://www.gnu.org/software/coreutils/chown>
or available locally via: info '(coreutils) chown invocation'
```

La Commande chgrp

Le même cas de figure s'applique au groupe :

```
# chgrp root tux.jpg [Entrée]
```

affectera le fichier au groupe root :

```
root@debian9:/home/trainee# chgrp root tux.jpg
root@debian9:/home/trainee# ls -l | grep tux.jpg
-rw-r--r-- 1 root    root       0 avril  4 13:54 tux.jpg
```

Rappel : Seul root peut changer le propriétaire d'un fichier.

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.

Options de la Commande

Les options de cette commande sont :

```
root@debian9:/home/trainee# ls -l | grep tux.jpg
-rw-r--r-- 1 root    root       0 avril  4 13:54 tux.jpg
root@debian9:/home/trainee# chgrp --help
Utilisation : chgrp [OPTION]... GROUPE FICHIER...
         ou : chgrp [OPTION]... --reference=FICHIER_R FICHIER...
Modifier le groupe de chaque FICHIER en GROUPE.
Avec --reference, modifier le groupe de chaque FICHIER à celui de FICHIER_R.

  -c, --changes          comme --verbose, mais seulement en cas de modification
  -f, --silent, --quiet  supprimer la plupart des messages d'erreur
  -v, --verbose          afficher un diagnostic pour chaque fichier traité
      --dereference      affecter le référent de chaque lien symbolique (par
                           défaut), au lieu du lien symbolique lui-même
  -h, --no-dereference   affecter les liens symboliques au lieu des fichiers
                           référencés
                           (seulement utile sur les systèmes permettant de
                           modifier le propriétaire d'un lien symbolique)
      --no-preserve-root ne pas traiter « / » de manière spéciale (par défaut)
      --preserve-root    bloquer le traitement récursif sur « / »
```

```
      --reference=FICHIER_R  utiliser le groupe de FICHIER_R au lieu d'indiquer
                               une valeur de GROUPE
  -R, --recursive        opérer récursivement sur les fichiers et répertoires

Les options suivantes modifient la façon de parcourir la hiérarchie lorsque
l'option -R est aussi indiquée. 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, le parcourir
  -L                     parcourir tous les liens symboliques menant à un
                           répertoire
  -P                     ne parcourir aucun lien symbolique (par défaut)

      --help     afficher l'aide et quitter
      --version  afficher des informations de version et quitter

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

Aide en ligne de GNU coreutils : <http://www.gnu.org/software/coreutils/>
Signalez les problèmes de traduction de « chgrp » à : <traduc@traduc.org>
Full documentation at: <http://www.gnu.org/software/coreutils/chgrp>
or available locally via: info '(coreutils) chgrp invocation'
```

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@debian9:/home/trainee# ls -l /etc/passwd /usr/bin/passwd
-rw-r--r-- 1 root root  1962 janv. 22 13:39 /etc/passwd
-rwsr-xr-x 1 root root 59680 mai   17  2017 /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

Afin d'identifier les exécutables ayant le SGID ou SUID bit, utilisez la commande suivante :

```
# find / -type f \( -perm -4000 -o -perm -2000 \) -exec ls {} \; [Entrée]
```

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@debian9:/home/trainee# cd /tmp
root@debian9:/tmp# mkdir inherit
root@debian9:/tmp# chown root:trainee inherit
root@debian9:/tmp# chmod g+s inherit
root@debian9:/tmp# touch inherit/test.txt
root@debian9:/tmp# mkdir inherit/testrep
root@debian9:/tmp# cd inherit; ls -l
total 4
drwxr-sr-x 2 root trainee 4096 avril  4 14:09 testrep
-rw-r--r-- 1 root trainee    0 avril  4 14:09 test.txt
```

Important : Notez que malgré le fait que root a créé les deux objets, ceux-ci ne sont pas associés avec le groupe **root** mais avec le groupe **trainee**, le groupe du répertoire parent (inherit). Notez aussi que le système a posé le drapeau d'héritage sur le sous-répertoire **testrep**.

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@debian9:/tmp/inherit# mkdir /tmp/repertoire_public; cd /tmp; chmod o+t repertoire_public
root@debian9:/tmp# ls -l | grep repertoire_public
drwxr-xr-t 2 root    root    4096 avril  4 14:10 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.

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

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

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

```
root@debian9:/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--
```

Positionner des ACL sur un Fichier

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]
```

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

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

Vous obtiendrez un résultat similaire à celui-ci :

```
root@debian9:/tmp# setfacl --set u::rwx,g::rx,o::-,u:trainee:rw /home/trainee/tux.jpg
root@debian9:/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::---
```

Important - Veuillez noter l'apparition de la ligne **mask**. Le mask indique les permissions maximales qui peuvent être acccordées à un utilisateur ou un groupe tiers.

Positionner des ACL sur un Répertoire

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@debian9:/tmp# mkdir /home/trainee/rep1
root@debian9:/tmp# setfacl --set d:u::r,d:g::-,d:o::- /home/trainee/rep1
root@debian9:/tmp# touch /home/trainee/rep1/fichier1
root@debian9:/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@debian9:/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]
```

Options de la Commande getfacl

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

```
root@debian9:/tmp# getfacl --help
getfacl 2.2.52 -- 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
```

Les Options de la Commande setfacl

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

```
root@debian9:/tmp# setfacl --help
setfacl 2.2.52 -- 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
```

Les Attributs Etendus

Les attributs s'ajoutent aux caractéristiques classiques d'un fichier dans un système de fichiers Ext2/Ext3/Ext4, JFS, ReiserFS, XFS et Btrfs.

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

Important - Un fichier synchrone et un répertoire synchrone impliquent que les modifications seront immédiatement inscrites sur disque.

Les commandes associées avec les attributs sont :

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

Préparation

Pour mieux comprendre, créez le répertoire **/tmp/attributs/rep** les fichier **fichier** et **rep/fichier1** :

```
root@debian9:/tmp# mkdir -p attributs/rep
root@debian9:/tmp# touch attributs/fichier
root@debian9:/tmp# touch attributs/rep/fichier1
```

Modifier les Attributs

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

```
root@debian9:/tmp# chattr +i -R attributs/
```

Visualiser les Attributs

Visualisez les attributs de l'arborescence **attributs** :

```
root@debian9:/tmp# lsattr -R attributs
```

```
----i---------e---- attributs/fichier
----i---------e---- attributs/rep

attributs/rep:
----i---------e---- attributs/rep/fichier1
```

Important - Notez que l'attribut **e** sous Ext4 indique l'utilisation des **Extents**. Cet attribut ne peut pas être enlever avec la commande **chattr**. Les Extents seront couverts dans le cours **Gestion des Disques, des Systèmes de Fichiers et le Swap**.

Résultat de la mise en place des Attributs

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

```
root@debian9:/tmp# cd attributs; mv /tmp/attributs/fichier /tmp/attributs/rep/fichier
mv: impossible de déplacer '/tmp/attributs/fichier' vers '/tmp/attributs/rep/fichier': Opération non permise
```

Options de la Commande chattr

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

```
root@debian9:/tmp/attributs# chattr --help
Utilisation : chattr [-pRVf] [-+=aAcCdDeijPsStTu] [-v version] fichiers...
```

Options de la Commande lsattr

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

```
root@debian9:/tmp/attributs# lsattr --help
lsattr : option invalide -- '-'
Utilisation : lsattr [-RVadlpv] [fichiers...]
```

<html>

</html>