

Dernière mise-à-jour : 2020/01/30 03:28

LSF108 - Gestion des Droits

Dans sa conception de base, Linux utilise une approche sécurité de type **DAC**. 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	<i>Discretionary Access Control</i>	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**:

```
$ touch tux.jpg [Entrée]
```

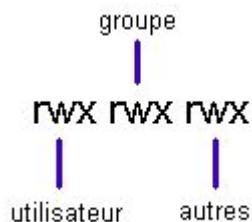
```
trainee@SUSE12sp1:~> pwd
/home/trainee
trainee@SUSE12sp1:~> touch tux.jpg
trainee@SUSE12sp1:~> ls -l | grep tux
-rw-r--r-- 1 trainee users 0 8 oct. 11:56 tux.jpg
```



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 accessibles 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 <i>catégories</i> (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@SUSE12sp1:~> chmod o+w tux.jpg
trainee@SUSE12sp1:~> ls -l | grep tux
-rw-r--rw- 1 trainee users 0 8 oct. 11:56 tux.jpg
```

Tandis que :

```
$ chmod ug-w tux.jpg [Entrée]
```

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

```
trainee@SUSE12sp1:~> chmod ug-w tux.jpg
trainee@SUSE12sp1:~> ls -l | grep tux
-r--r--rw- 1 trainee users 0 8 oct. 11:56 tux.jpg
```



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

Mode Octal

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

r	w	x	r	w	x	r	w	x
400	200	100	40	20	10	4	2	1
<hr/>			<hr/>			<hr/>		
Utilisateur			Group			Other		



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@SUSE12sp1:~> chmod 644 tux.jpg
trainee@SUSE12sp1:~> ls -l | grep tux
-rw-r--r-- 1 trainee users 0 8 oct. 11:56 tux.jpg
```



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

Options de la Commande

Les options de cette commande sont :

```
trainee@SUSE12sp1:~> chmod --help
Utilisation : chmod [OPTION]... MODE[,MODE]... FILE...
    ou : chmod [OPTION]... OCTAL-MODE FILE
    ou : chmod [OPTION]... --reference=FILE FILE
Modifier le mode de chaque FILE en MODE.
Avec --reference, modifier le mode de chaque FILE à celui de FILE.
-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=RFILE  utiliser le mode de RFILE au lieu d'indiquer une
                   valeur GROUP
-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>

For complete documentation, run: 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 RHEL/CentOS est différente pour un utilisateur normal et pour root :

```
trainee@SUSE12sp1:~> umask
0022
trainee@SUSE12sp1:~> su -
Mot de passe :
SLES12SP1:~ # umask
0022
SLES12SP1:~ # exit
logout
trainee@SUSE12sp1:~>
```

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

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

```
trainee@SUSE12sp1:~> umask 044
trainee@SUSE12sp1:~> touch tux1.jpg
trainee@SUSE12sp1:~> ls -l | grep tux
-rw--w--w- 1 trainee users 0 8 oct. 12:00 tux1.jpg
-rw-r--r-- 1 trainee users 0 8 oct. 11:56 tux.jpg
trainee@SUSE12sp1:~> umask 022
trainee@SUSE12sp1:~> umask
0022
```

Options de la Commande

Les options de cette commande sont :

```
trainee@SUSE12sp1:~> 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.

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@SUSE12sp1:~> su -
```

```
Mot de passe :
```

```
SLES12SP1:~ # cd /home/trainee
```

```
SLES12SP1:/home/trainee # chown root tux.jpg
```

```
SLES12SP1:/home/trainee # ls -l | grep tux
```

```
-rw-r--r-- 1 root users 0 Oct 8 11:56 tux.jpg
```

```
-rw--w--w- 1 trainee users 0 Oct 8 12:00 tux1.jpg
```

Options de la Commande

Les options de cette commande sont :

```
trainee@SUSE12sp1:~> chown --help
Utilisation : chown [OPTION]... [OWNER][:GROUP] FILE...
               ou : chown [OPTION]... --reference=RFILE FILE...
Modifier le propriétaire ou le groupe de chaque FILE en OWNER ou GROUP.
Avec --reference, modifier le propriétaire et le groupe de chaque FILE à
ceux de RFILE.

-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=CURRENT_OWNER:CURRENT_GROUP
                      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=RFILE   utiliser les propriétaires et groupe de RFILE au lieu
                  d'indiquer des valeurs OWNER:GROUP
-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 si l n'est pas indiqué, mais modifié en groupe de connexion s'il est sous-entendu par un « : » suivant un OWNER (propriétaire) symbolique.

Le OWNER et le GROUP 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>

For complete documentation, run: 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 :

```
SLES12SP1:/home/trainee # chgrp root tux.jpg
SLES12SP1:/home/trainee # ls -l | grep tux
-rw-r--r-- 1 root      root      0 Oct  8 11:56 tux.jpg
-rw--w--w- 1 trainee   users     0 Oct  8 12:00 tux1.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 :

```
SLES12SP1:/home/trainee # chgrp --help
Usage: chgrp [OPTION]... GROUP FILE...
      or: chgrp [OPTION]... --reference=RFILE FILE...
Change the group of each FILE to GROUP.
With --reference, change the group of each FILE to that of RFILE.

      -c, --changes          like verbose but report only when a change is made
      -f, --silent, --quiet  suppress most error messages
      -v, --verbose          output a diagnostic for every file processed
      --dereference          affect the referent of each symbolic link (this is
                            the default), rather than the symbolic link itself
      -h, --no-dereference   affect symbolic links instead of any referenced file
```

```
        (useful only on systems that can change the
         ownership of a symlink)
--no-preserve-root  do not treat '/' specially (the default)
--preserve-root     fail to operate recursively on '/'
--reference=RFILE  use RFILE's group rather than specifying a
                   GROUP value
-R, --recursive     operate on files and directories recursively
```

The following options modify how a hierarchy is traversed when the -R option is also specified. If more than one is specified, only the final one takes effect.

```
-H                  if a command line argument is a symbolic link
                   to a directory, traverse it
-L                  traverse every symbolic link to a directory
                   encountered
-P                  do not traverse any symbolic links (default)

--help    display this help and exit
--version output version information and exit
```

Examples:

```
chgrp staff /u      Change the group of /u to "staff".
chgrp -hR staff /u  Change the group of /u and subfiles to "staff".
```

GNU coreutils online help: <<http://www.gnu.org/software/coreutils/>>
Report chgrp translation bugs to <<http://translationproject.org/team/>>
For complete documentation, run: 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.

```
SLES12SP1:/home/trainee # ls -l /etc/passwd /usr/bin/passwd
-rw-r--r-- 1 root root 1776 Oct  8 07:12 /etc/passwd
-rwsr-xr-x 1 root shadow 51200 Aug 21 2015 /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 :

```
SLES12SP1:/home/trainee # cd /tmp
SLES12SP1:/tmp # mkdir inherit
SLES12SP1:/tmp # chown root:trainee inherit
chown: invalid group: 'root:trainee'
SLES12SP1:/tmp # chown root:users inherit
SLES12SP1:/tmp # chmod g+s inherit
SLES12SP1:/tmp # touch inherit/test.txt
SLES12SP1:/tmp # mkdir inherit/testrep
SLES12SP1:/tmp # cd inherit; ls -l
total 0
-rw-r--r-- 1 root users 0 Oct  8 12:08 test.txt
drwxr-sr-x 1 root users 0 Oct  8 12:08 testrep
```



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 **users**, 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 :

```
SLES12SP1:/tmp/inherit # mkdir /tmp/public_directory; cd /tmp; chmod o+t public_directory
SLES12SP1:/tmp # ls -l | grep public_directory
drwxr-xr-t 1 root      root      0 Oct  8 12:15 public_directory
```

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 :

```
SLES12SP1:/tmp # getfacl /home/trainee/tux.jpg
getfacl: Removing leading '/' from absolute path names
# 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]
```

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

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

Vous obtiendrez un résultat similaire à celui-ci :

```
SLES12SP1:/tmp # setfacl --set u::rwx,g::rx,o::-,u:trainee:rw /home/trainee/tux.jpg
SLES12SP1:/tmp # getfacl /home/trainee/tux.jpg
getfacl: Removing leading '/' from absolute path names
# 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 accordées à un utilisateur ou un groupe tiers.

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 :

```
SLES12SP1:/tmp # mkdir /home/trainee/rep1
SLES12SP1:/tmp # setfacl --set d:u::r,d:g::-,d:o::-- /home/trainee/rep1
SLES12SP1:/tmp # touch /home/trainee/rep1/fichier1
SLES12SP1:/tmp # getfacl /home/trainee/rep1
```

```
getfacl: Removing leading '/' from absolute path names
# file: home/trainee/rep1
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
default:user::r--
default:group::---
default:other::---
```

```
SLES12SP1:/tmp # getfacl /home/trainee/rep1/fichier1
getfacl: Removing leading '/' from absolute path names
# 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 des Commandes

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

```
SLES12SP1:/tmp # getfacl --help
getfacl 2.2.52 -- get file access control lists
Usage: getfacl [-aceEsRLPtpndvh] file ...
-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** sont :

```
SLES12SP1:/tmp # setfacl --help
setfacl 2.2.52 -- set file access control lists
Usage: setfacl [-bkndRLP] { -m|-M|-x|-X ... } 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
--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	don't recalculate the effective rights mask
-d, --default	operations apply to the default ACL
-R, --recursive	recurse into subdirectories
-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

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 être 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

Commande	description
lsattr	Visualise les attributs

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

```
SLES12SP1:/tmp # cd /root
SLES12SP1:~ # mkdir -p attributs/rep
```

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

```
SLES12SP1:~ # touch attributs/fichier
SLES12SP1:~ # touch attributs/rep/fichier1
```

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

```
SLES12SP1:~ # chattr +i -R attributs/
```

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

```
SLES12SP1:~ # lsattr -R attributs
----i----- attributs/rep

attributs/rep:
----i----- attributs/rep/fichier1

----i----- attributs/fichier
```



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**.

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

```
SLES12SP1:~ # cd attributs; mv /root/attributs/fichier /root/attributs/rep/fichier
mv: cannot move '/root/attributs/fichier' to '/root/attributs/rep/fichier': Permission denied
```

Options des Commandes

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

```
SLES12SP1:~/attributs # chattr --help
Usage: chattr [-RVf] [-+=aAcCdDeijsStTu] [-v version] files...
```

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

```
SLES12SP1:~/attributs # lsattr --help
lsattr: invalid option -- '-'
Usage: lsattr [-RVadlv] [files...]
```

<html>

Copyright © 2004-2017 I2TCH LIMITED.

</html>

From:

<https://www.ittraining.team/> - **www.ittraining.team**



Permanent link:

<https://www.ittraining.team/doku.php?id=elearning:workbooks:opensuse:11:junior:l108>

Last update: **2020/01/30 03:28**