Dernière mise-à-jour : 2020/01/30 03:27

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 Ext2/Ext3/Ext4** :

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

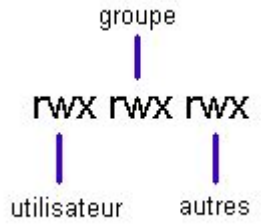
Dans votre répertoire personnel, créez un fichier tux.jpg grâce à la commande **touch**:

```
root@ubuntu:~# exit
déconnexion
trainee@ubuntu:~$ pwd
/home/trainee
trainee@ubuntu:~$ touch tux.jpg
trainee@ubuntu:~$ ls -l | grep tux
-rw-rw-r-- 1 trainee trainee    0 oct.   6 10:04 tux.jpg
```

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

Les options de cette commande sont :

```
trainee@ubuntu:~$ chmod --help
Utilisation : chmod [OPTION]... MODE[,MODE]... FILE...
         ou : chmod [OPTION]... OCTAL-MODE FILE
         ou : chmod [OPTION]... --reference=RFILE FILE
Modifier le mode de chaque FILE en MODE.
Avec --reference, modifier le mode de chaque FILE à celui 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
      --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]+ ».

Signalez les anomalies de « chmod » à <bug-coreutils@gnu.org>
Page d'accueil de « GNU coreutils » : http://www.gnu.org/software/coreutils/
Aide globale sur les logiciels GNU : <http://www.gnu.org/help/gethelp>
Signalez les problèmes de traduction de « chmod » à : <traduc@traduc.org>
Utilisez « info coreutils 'chmod invocation' » pour toute la documentation
```

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@ubuntu:~$ chmod o+w tux.jpg
trainee@ubuntu:~$ ls -l | grep tux
-rw-rw-rw- 1 trainee trainee    0 oct.   6 10:04 tux.jpg
```

Tandis que :

```
$ chmod ug-w tux.jpg [Entrée]
```

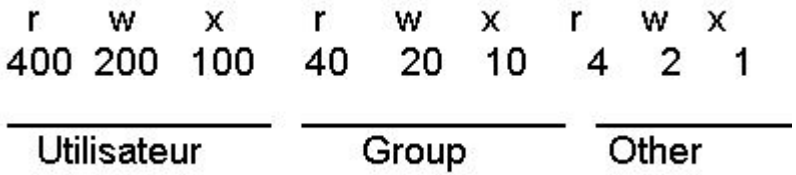
ôtera les droit d'accès en écriture pour l'utilisateur et le groupe :

```
trainee@ubuntu:~$ chmod ug-w tux.jpg
trainee@ubuntu:~$ ls -l | grep tux
-r--r--rw- 1 trainee trainee    0 oct.   6 10:04 tux.jpg
```

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 :



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@ubuntu:~$ chmod 644 tux.jpg
trainee@ubuntu:~$ ls -l | grep tux
-rw-r--r-- 1 trainee trainee    0 oct.   6 10:04 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

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@ubuntu:~$ 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, afficher sous une forme réutilisable comme une entrée
      -S    afficher 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.
```

La valeur par défaut de l'umask sous Ubuntu n'est pas identique pour un utilisateur normal et pour root :

```
trainee@ubuntu:~$ umask
0002
trainee@ubuntu:~$ sudo su -
[sudo] password for trainee:
root@ubuntu:~# umask
0022
root@ubuntu:~# exit
déconnexion
trainee@ubuntu:~$
```

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@ubuntu:~$ umask 044
```

```
trainee@ubuntu:~$ touch tux1.jpg
trainee@ubuntu:~$ ls -l | grep tux1.jpg
-rw--w--w- 1 trainee trainee    0 oct.   6 12:20 tux1.jpg
trainee@ubuntu:~$ umask 002
trainee@ubuntu:~$ umask
0002
```

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@ubuntu:~$ 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 sil 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 ».
```

```
Signalez les anomalies de « chown » à <bug-coreutils@gnu.org>
Page d'accueil de « GNU coreutils » : http://www.gnu.org/software/coreutils/
Aide globale sur les logiciels GNU : <http://www.gnu.org/help/gethelp>
Signalez les problèmes de traduction de « chown » à : <traduc@traduc.org>
Utilisez « info coreutils 'chown invocation' » pour toute la documentation
```

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@ubuntu:~$ sudo su -
[sudo] password for trainee:
root@ubuntu:~# cd /home/trainee/
root@ubuntu:/home/trainee# chown root tux.jpg
root@ubuntu:/home/trainee# ls -l | grep tux.jpg
-rw-r--r-- 1 root    trainee    0 oct.   6 10:04 tux.jpg
```

La Commande chrgp

Les options de cette commande sont détaillées ci-après :

```
root@ubuntu:/home/trainee# chgrp --help
Utilisation : chgrp [OPTION]... GROUP FILE...
         ou : chgrp [OPTION]... --reference=RFILE FILE...
Modifier le groupe de chaque FILE en GROUP.
Avec --reference, modifier le groupe de chaque FILE à celui 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)
      --no-preserve-root ne pas traiter « / » de manière spéciale (par défaut)
      --preserve-root    bloquer le traitement récursif sur « / »
      --reference=RFILE  utiliser le groupe de RFILE au lieu d'indiquer une
                         valeur 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

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 ».

Signalez les anomalies de « chgrp » à <bug-coreutils@gnu.org>
Page d'accueil de « GNU coreutils » : http://www.gnu.org/software/coreutils/
Aide globale sur les logiciels GNU : <http://www.gnu.org/help/gethelp>
Signalez les problèmes de traduction de « chgrp » à : <traduc@traduc.org>
Utilisez « info coreutils 'chgrp invocation' » pour toute la documentation
```

Le même cas de figure s’applique au groupe :

```
# chgrp root tux.jpg [Entrée]
```

affectera le fichier au groupe root :

```
root@ubuntu:/home/trainee# chgrp root tux.jpg
root@ubuntu:/home/trainee# ls -l | grep tux.jpg
-rw-r--r-- 1 root    root       0 oct.   6 10:04 tux.jpg
```

Seul root peut changer le propriétaire d'un fichier.

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.

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@ubuntu:/home/trainee# ls -l /etc/passwd /usr/bin/passwd
-rw-r--r-- 1 root root  2020 oct.   1 16:51 /etc/passwd
-rwsr-xr-x 1 root root 45420 févr. 17  2014 /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 :

```
root@ubuntu:/home/trainee# find / -type f \( -perm -4000 -o -perm -2000 \) -exec ls -l {} \; 2>/dev/null
-rwsr-xr-- 1 root dip 322968 janv. 22  2013 /usr/sbin/pppd
-rwsr-sr-x 1 libuuid libuuid 17996 juin   3 22:54 /usr/sbin/uuidd
-rwxr-sr-x 1 root utmp 14004 nov.  22  2013 /usr/lib/libvte-2.90-9/gnome-pty-helper
-rwxr-sr-x 1 root mail 13888 mai   15 21:34 /usr/lib/evolution/camel-lock-helper-1.2
-rwsr-xr-x 1 root root 17936 août  28 18:16 /usr/lib/i386-linux-gnu/oxide-qt/chrome-sandbox
-rwsr-xr-x 1 root root 9612 avril 12 12:44 /usr/lib/pt_chown
-rwsr-xr-- 1 root messagebus 329856 juil.  3 21:04 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwxr-sr-x 1 root utmp 9468 oct.   5  2012 /usr/lib/utempter/utempter
```

```
-rwsr-xr-x 1 root root 9804 févr. 11  2014 /usr/lib/policykit-1/polkit-agent-helper-1
-rwsr-xr-x 1 root root 5480 févr. 25  2014 /usr/lib/eject/dmcrypt-get-device
-rwsr-xr-x 1 root root 492972 mai   12 18:06 /usr/lib/openssh/ssh-keysign
-rwxr-sr-x 1 root tty 9576 déc.   5  2013 /usr/lib/mc/cons.saver
-rwxr-sr-x 1 root mail 9704 déc.   4  2012 /usr/bin/mail-lock
-rwxr-sr-x 1 root shadow 49420 févr. 17  2014 /usr/bin/chage
-rwxr-sr-x 1 root mail 13960 déc.   7  2013 /usr/bin/dotlockfile
-rwsr-xr-x 1 root root 72860 oct.  21  2013 /usr/bin/mtr
-rwsr-xr-x 1 root root 45420 févr. 17  2014 /usr/bin/passwd
-rwxr-sr-x 1 root ssh 329144 mai   12 18:06 /usr/bin/ssh-agent
-rwxr-sr-x 1 root mlocate 34452 juin  20  2013 /usr/bin/mlocate
-rwsr-xr-x 1 root root 18136 mai    7 23:51 /usr/bin/traceroute6.iputils
-rwsr-sr-x 1 root root 9532 janv. 29  2014 /usr/bin/X
-rwxr-sr-x 1 root crontab 34824 févr.  9  2013 /usr/bin/crontab
-rwsr-xr-x 1 root lpadmin 13672 juil. 18 22:58 /usr/bin/lppasswd
-rwxr-sr-x 1 root shadow 18208 févr. 17  2014 /usr/bin/expiry
-rwsr-xr-x 1 root root 30984 févr. 17  2014 /usr/bin/newgrp
-rwsr-xr-x 1 root root 66252 févr. 17  2014 /usr/bin/gpasswd
-rwxr-sr-x 1 root mail 9704 déc.   4  2012 /usr/bin/mail-unlock
-rwsr-xr-x 1 root root 156708 févr. 10  2014 /usr/bin/sudo
-rwxr-sr-x 1 root tty 18056 juin   3 22:54 /usr/bin/wall
-rwsr-xr-x 1 root root 35916 févr. 17  2014 /usr/bin/chsh
-rwsr-xr-x 1 root root 44620 févr. 17  2014 /usr/bin/chfn
-rwsr-xr-x 1 root root 18168 févr. 11  2014 /usr/bin/pkexec
-rwxr-sr-x 1 root mail 9704 déc.   4  2012 /usr/bin/mail-touchlock
-rwxr-sr-x 1 root tty 9748 juin   4  2013 /usr/bin/bsd-write
-rwxr-sr-x 1 root shadow 30432 janv. 31  2014 /sbin/unix_chkpwd
-rwsr-xr-x 1 root root 88752 juin   3 22:54 /bin/mount
-rwsr-xr-x 1 root root 30112 déc.  16  2013 /bin/fusermount
-rwsr-xr-x 1 root root 43316 mai    7 23:51 /bin/ping6
-rwsr-xr-x 1 root root 67704 juin   3 22:54 /bin/umount
-rwsr-xr-x 1 root root 35300 févr. 17  2014 /bin/su
-rwsr-xr-x 1 root root 38932 mai    7 23:51 /bin/ping
```

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@ubuntu:/home/trainee# cd /tmp
root@ubuntu:/tmp# mkdir inherit
root@ubuntu:/tmp# chown root:trainee inherit
root@ubuntu:/tmp# chmod g+s inherit
root@ubuntu:/tmp# touch inherit/test.txt
root@ubuntu:/tmp# mkdir inherit/testrep
root@ubuntu:/tmp# cd inherit; ls -l
total 4
drwxr-sr-x 2 root trainee 4096 oct.   6 13:45 testrep
-rw-r--r-- 1 root trainee    0 oct.   6 13:45 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@ubuntu:/tmp/inherit# mkdir /tmp/repertoire_public; cd /tmp; chmod o+t repertoire_public
root@ubuntu:/tmp# ls -l | grep repertoire_public
drwxr-xr-t 2 root    root    4096 oct.   6 13:46 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 Ubuntu, le paquet **acl** est déjà installé :

```
root@ubuntu:/tmp# apt-get install acl
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
acl est déjà la plus récente version disponible.
0 mis à jour, 0 nouvellement installés, 0 à enlever et 161 non mis à jour.
```

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

```
# /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>
# / was on /dev/sda1 during installation
UUID=70eb8bc5-1759-433d-9797-9342a7b82cb2 /               ext4    errors=remount-ro,acl 0       1
# swap was on /dev/sda5 during installation
UUID=a8b7b615-a3de-410a-8184-2acf798f6144 none            swap    sw              0       0
```

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

```
root@ubuntu:/tmp# mount -o remount /
root@ubuntu:/tmp# mount | grep acl
/dev/sda1 on / type ext4 (rw,errors=remount-ro,acl)
```

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

```
root@ubuntu:/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--
```

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

```
root@ubuntu:/tmp# getfacl --help
getfacl 2.2.52 -- obtenir les listes de contrôle d'accès du fichier
Utilisation : getfacl [-aceEsRLPtpndvh] fichier...
  -a,  --access affiche la liste de contrôle d’accès du fichier seule
```

```
  -d, --default affiche la liste de contrôle d’accès par défaut seule
  -c, --omit-header supprime les commentaires à l'affichage
  -e, --all-effective affiche les droits effectifs
  -E, --no-effective affiche les droits non effectifs
  -s, --skip-base ignore les fichiers qui n'ont que les entrées de la base
  -R, --recursive traitement récursif des sous-dossiers
  -L, --logical parcours logique, suit les liens symboliques
  -P, --physical parcours physique, ne suit les liens symboliques
  -t, --tabular affiche les colonnes séparées par des tabulations
  -n, --numeric affiche les identifiants numériques d'utilisateur et de groupes
  -p, --absolute-names préserve le caractère « / » en début de chemin
  -v, --version           affiche la version et quitte
  -h, --help              affiche ce texte d'aide
```

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

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

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

```
root@ubuntu:/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 définit l'ACL du (des) fichier(s), remplaçant ainsi l'ACL courant
      --set-file=file lit les entrées ACL à définir depuis un fichier
      --mask recalcule les masques de droits en vigueur
  -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           affiche la version et quitte
  -h, --help              affiche ce texte d'aide
```

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

```
root@ubuntu:/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, **trainee** a maintenant les permissions rw.

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

```
         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).
```

Les Attributs Ext2/Ext3/Ext4

Les attributs s'ajoutent aux caractéristiques classiques d'un fichier dans 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

Attribut	Description
S	Fichier synchrone
A	La date et l'heure de dernier accès ne seront pas mises à jour

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

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

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

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

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

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

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

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

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

```
attributs/rep:
```

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

```
root@ubuntu:/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
```

<html>

</html>
