

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

LSF101 - Système de Fichiers

Linux File Hierarchy System

Le système de fichiers de Linux est organisé autour d'une arborescence unique ayant un point de départ appelé la **racine**, représenté par le caractère `/`. En dessous de cette racine se trouvent des répertoires contenant fichiers et sous-répertoires. L'organisation des répertoires est conforme à un standard, appelé le **Linux File Hierarchy System**.

Arborescence

Cette organisation donne l'arborescence ci-dessous :

```
SLES12SP1:~ # cd /; ls -l
total 0
drwxr-x---  1 root root  42 May  3 14:02 .snapshots
drwxr-xr-x  1 root root 1810 May  1 15:31 bin
drwxr-xr-x  1 root root  642 May  2 15:58 boot
drwxr-xr-x 16 root root 3620 Sep 30 16:37 dev
drwxr-xr-x  1 root root 5200 Sep 30 16:37 etc
drwxr-xr-x  1 root root   14 May  1 15:58 home
drwxr-xr-x  1 root root 3120 May  1 15:21 lib
drwxr-xr-x  1 root root 5024 May  1 15:18 lib64
drwxr-xr-x  1 root root    0 Sep 21  2014 mnt
drwxr-xr-x  1 root root   44 May  3 13:59 opt
dr-xr-xr-x 105 root root    0 Sep 30 10:23 proc
drwx-----  1 root root  274 May  3 14:05 root
drwxr-xr-x 25 root root  560 Sep 30 16:37 run
drwxr-xr-x  1 root root 5100 May  1 15:51 sbin
```

```
drwxr-xr-x  1 root root    0 Sep 21  2014 selinux
drwxr-xr-x  1 root root   12 May  1 14:55 srv
dr-xr-xr-x 12 root root    0 Sep 30 10:23 sys
drwxrwxrwt  1 root root 802 Sep 30 16:37 tmp
drwxr-xr-x  1 root root  130 May  1 14:55 usr
drwxr-xr-x  1 root root  108 May  1 15:26 var
```

Directory	Contenu
/bin	Une abréviation de binary ou binaires. Il contient des programmes tels ls.
/boot	Contient les fichiers nécessaires au démarrage du système.
/dev	Contient les nœuds utilisés pour accéder à tout type de matériel tel /dev/fd0 pour le lecteur de disquette. C'est le binaire <i>udev</i> qui se charge de créer et supprimer d'une manière dynamique les nœuds.
/etc	Contient des fichiers de configuration tels passwd pour les mots de passe et fstab qui est la liste des systèmes de fichiers à monter lors du démarrage du système.
/home	Contient les répertoires de chaque utilisateur sauf l'utilisateur root.
/lib	Contient les bibliothèques 32 bits communes utilisées par les programmes ainsi que les modules.
/lib64	Contient les bibliothèques 64 bits communes utilisées par les programmes ainsi que les modules.
/lost+found	Contient des fragments de fichiers endommagés et retrouvés par la commande <i>fsck</i> .
/media	Contient des répertoires pour chaque système de fichiers monté (accessible au système linux) tels floppy, cdrom etc.
/mnt	Contient des répertoires pour chaque système de fichiers monté temporairement par root.
/opt	Contient des applications optionnelles.
/proc	Contient un système de fichiers virtuel qui extrait de la mémoire les informations en cours de traitement. Le contenu des fichiers est créé dynamiquement lors de la consultation. Seul root peut consulter la totalité des informations dans le répertoire /proc.
/root	Le home de root, l'administrateur système
/run	Remplace le répertoire /var/run . Notez que sous SLES 12, /var/run est un lien symbolique pointant vers /run .
/sbin	Contient des binaires, donc programmes, pour l'administration du système local.
/selinux	Contient des fichiers propres à l'implémentation de SELINUX.
/srv	Contient des données pour les services hébergés par le système tels ftp, bases de données, web etc.
/sys	Contient un système de fichiers virtuel dont le rôle est de décrire le matériel pour udev.
/tmp	Contient des fichiers temporaires créés par des programmes.
/usr	contient des commandes des utilisateurs dans /usr/bin, les HOWTO dans /usr/share/doc, les manuels dans /usr/share/man ainsi que d'autres entrées majeures.

Directory	Contenu
/var	contient des fichiers de taille variable tels les journaux et les spooleurs d'impression.

Types de Fichiers

Il existe trois types majeurs de fichier sous le système Linux :

- les fichiers normaux (ordinary files),
- les répertoires (directories),
- les fichiers spéciaux (special files ou Devices).

Notez que :

- Les fichiers normaux sont des fichiers textes, des tableaux ou des exécutables.
- La longueur du nom de fichier, y compris son extension, est limitée à 255 caractères.
- Il y a une distinction entre les majuscules et les minuscules.
- Si le nom d'un fichier commence par un ., le fichier devient caché.

La Commande mount

Pour que Linux soit informé de la présence d'un système de fichiers, ce système doit être monté. Pour monter un système de fichiers, on utilise la commande **mount** :

```
# mount /dev/<fichier_spécial> /mnt/<répertoire_cible>
```

ou **/dev/<fichier_spécial>** est le périphérique à monter et **/mnt/<répertoire_cible>** est le répertoire qui servira comme «fenêtre» pour visionner le contenu du système de fichiers. Ce répertoire doit impérativement exister avant d'essayer de monter le système de fichiers.

Dans le cas où la commande **mount** est utilisée sans options, le système retourne une liste de tous les systèmes de fichiers actuellement montés :

```
SLES12SP1:~ # mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
devtmpfs on /dev type devtmpfs (rw,nosuid,size=1931968k,nr_inodes=482992,mode=755)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,mode=755)
tmpfs on /sys/fs/cgroup type tmpfs (rw,nosuid,nodev,noexec,mode=755)
cgroup on /sys/fs/cgroup/systemd type cgroup
(rw,nosuid,nodev,noexec,relatime,xattr,release_agent=/usr/lib/systemd/systemd-cgroups-agent,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpuacct,cpu)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)
/dev/sda2 on / type btrfs (rw,relatime,space_cache,subvolid=259,subvol=@/.snapshots/1/snapshot)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs
(rw,relatime,fd=31,prgrp=1,timeout=300,minproto=5,maxproto=5,direct)
mqueue on /dev/mqueue type mqueue (rw,relatime)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime)
/dev/sda2 on /.snapshots type btrfs (rw,relatime,space_cache,subvolid=258,subvol=@/.snapshots)
/dev/sda2 on /var/lib/mailman type btrfs (rw,relatime,space_cache,subvolid=269,subvol=@/var/lib/mailman)
/dev/sda2 on /var/lib/mariadb type btrfs (rw,relatime,space_cache,subvolid=270,subvol=@/var/lib/mariadb)
/dev/sda2 on /var/log type btrfs (rw,relatime,space_cache,subvolid=274,subvol=@/var/log)
/dev/sda2 on /tmp type btrfs (rw,relatime,space_cache,subvolid=265,subvol=@/tmp)
/dev/sda2 on /var/spool type btrfs (rw,relatime,space_cache,subvolid=276,subvol=@/var/spool)
/dev/sda2 on /var/lib/named type btrfs (rw,relatime,space_cache,subvolid=272,subvol=@/var/lib/named)
/dev/sda2 on /srv type btrfs (rw,relatime,space_cache,subvolid=264,subvol=@/srv)
```

```
/dev/sda2 on /usr/local type btrfs (rw,relatime,space_cache,subvolid=266,subvol=/@/usr/local)
/dev/sda2 on /var/opt type btrfs (rw,relatime,space_cache,subvolid=275,subvol=/@/var/opt)
/dev/sda2 on /var/lib/pgsql type btrfs (rw,relatime,space_cache,subvolid=273,subvol=/@/var/lib/pgsql)
/dev/sda2 on /opt type btrfs (rw,relatime,space_cache,subvolid=263,subvol=/@/opt)
/dev/sda2 on /var/tmp type btrfs (rw,relatime,space_cache,subvolid=277,subvol=/@/var/tmp)
/dev/sda2 on /var/lib/mysql type btrfs (rw,relatime,space_cache,subvolid=271,subvol=/@/var/lib/mysql)
/dev/sda2 on /var/lib/libvirt/images type btrfs
(rw,relatime,space_cache,subvolid=268,subvol=/@/var/lib/libvirt/images)
/dev/sda2 on /var/crash type btrfs (rw,relatime,space_cache,subvolid=267,subvol=/@/var/crash)
/dev/sda2 on /home type btrfs (rw,relatime,space_cache,subvolid=262,subvol=/@/home)
/dev/sda2 on /boot/grub2/x86_64-efi type btrfs (rw,relatime,space_cache,subvolid=261,subvol=/@/boot/grub2/x86_64-efi)
/dev/sda2 on /boot/grub2/i386-pc type btrfs (rw,relatime,space_cache,subvolid=260,subvol=/@/boot/grub2/i386-pc)
```



Important : Notez que sous SLES 12, le système de fichier par défaut est **btrfs**. Veuillez consulter le cours **Gestion des Disques, des Systèmes de Fichiers et du Swap** pour plus d'informations concernant les systèmes de fichiers.

Options de la commande

Les options de la commande mount sont :

```
SLES12SP1:~ # mount --help
```

Usage:

```
mount [-lhV]
mount -a [options]
mount [options] [--source] <source> | [--target] <directory>
mount [options] <source> <directory>
mount <operation> <mountpoint> [<target>]
```

Options:

-a, --all	mount all filesystems mentioned in fstab
-c, --no-canonicalize	don't canonicalize paths
-f, --fake	dry run; skip the mount(2) syscall
-F, --fork	fork off for each device (use with -a)
-T, --fstab <path>	alternative file to /etc/fstab
-h, --help	display this help text and exit
-i, --internal-only	don't call the mount.<type> helpers
-l, --show-labels	lists all mounts with LABELs
-n, --no-mtab	don't write to /etc/mtab
-o, --options <list>	comma-separated list of mount options
-O, --test-opts <list>	limit the set of filesystems (use with -a)
-r, --read-only	mount the filesystem read-only (same as -o ro)
-t, --types <list>	limit the set of filesystem types
--source <src>	explicitly specifies source (path, label, uuid)
--target <target>	explicitly specifies mountpoint
-v, --verbose	say what is being done
-V, --version	display version information and exit
-w, --rw, --read-write	mount the filesystem read-write (default)

-h, --help	display this help and exit
-V, --version	output version information and exit

Source:

-L, --label <label>	synonym for LABEL=<label>
-U, --uuid <uuid>	synonym for UUID=<uuid>
LABEL=<label>	specifies device by filesystem label
UUID=<uuid>	specifies device by filesystem UUID
PARTLABEL=<label>	specifies device by partition label
PARTUUID=<uuid>	specifies device by partition UUID
<device>	specifies device by path
<directory>	mountpoint for bind mounts (see --bind/rbind)
<file>	regular file for loopdev setup

Operations:

-B, --bind	mount a subtree somewhere else (same as -o bind)
-M, --move	move a subtree to some other place
-R, --rbind	mount a subtree and all submounts somewhere else
--make-shared	mark a subtree as shared
--make-slave	mark a subtree as slave
--make-private	mark a subtree as private
--make-unbindable	mark a subtree as unbindable
--make-rshared	recursively mark a whole subtree as shared
--make-rslave	recursively mark a whole subtree as slave
--make-rprivate	recursively mark a whole subtree as private
--make-runbindable	recursively mark a whole subtree as unbindable

For more details see `mount(8)`.

Le Fichier `/etc/fstab`

Dans le cas où la commande **mount** est utilisée avec l'option **-a**, tous les systèmes de fichiers mentionnés dans un fichier spécial dénommé **/etc/fstab** seront montés en même temps.

```
SLES12SP1:~ # cat /etc/fstab
UUID=db743358-c2d6-47f6-97d7-e7a9c650f0c5 swap swap defaults 0 0
UUID=6b7e374a-ae42-4f93-b6aa-d288dfbbb74b / btrfs defaults 0 0
UUID=6b7e374a-ae42-4f93-b6aa-d288dfbbb74b /boot/grub2/i386-pc btrfs subvol=@/boot/grub2/i386-pc 0 0
UUID=6b7e374a-ae42-4f93-b6aa-d288dfbbb74b /boot/grub2/x86_64-efi btrfs subvol=@/boot/grub2/x86_64-efi 0 0
UUID=6b7e374a-ae42-4f93-b6aa-d288dfbbb74b /home btrfs subvol=@/home 0 0
UUID=6b7e374a-ae42-4f93-b6aa-d288dfbbb74b /opt btrfs subvol=@/opt 0 0
UUID=6b7e374a-ae42-4f93-b6aa-d288dfbbb74b /srv btrfs subvol=@/srv 0 0
UUID=6b7e374a-ae42-4f93-b6aa-d288dfbbb74b /tmp btrfs subvol=@/tmp 0 0
UUID=6b7e374a-ae42-4f93-b6aa-d288dfbbb74b /usr/local btrfs subvol=@/usr/local 0 0
UUID=6b7e374a-ae42-4f93-b6aa-d288dfbbb74b /var/crash btrfs subvol=@/var/crash 0 0
UUID=6b7e374a-ae42-4f93-b6aa-d288dfbbb74b /var/lib/libvirt/images btrfs subvol=@/var/lib/libvirt/images 0 0
```

```

UUID=6b7e374a-ae42-4f93-b6aa-d288dfbbb74b /var/lib/mailman btrfs subvol=@/var/lib/mailman 0 0
UUID=6b7e374a-ae42-4f93-b6aa-d288dfbbb74b /var/lib/mariadb btrfs subvol=@/var/lib/mariadb 0 0
UUID=6b7e374a-ae42-4f93-b6aa-d288dfbbb74b /var/lib/mysql btrfs subvol=@/var/lib/mysql 0 0
UUID=6b7e374a-ae42-4f93-b6aa-d288dfbbb74b /var/lib/named btrfs subvol=@/var/lib/named 0 0
UUID=6b7e374a-ae42-4f93-b6aa-d288dfbbb74b /var/lib/pgsql btrfs subvol=@/var/lib/pgsql 0 0
UUID=6b7e374a-ae42-4f93-b6aa-d288dfbbb74b /var/log btrfs subvol=@/var/log 0 0
UUID=6b7e374a-ae42-4f93-b6aa-d288dfbbb74b /var/opt btrfs subvol=@/var/opt 0 0
UUID=6b7e374a-ae42-4f93-b6aa-d288dfbbb74b /var/spool btrfs subvol=@/var/spool 0 0
UUID=6b7e374a-ae42-4f93-b6aa-d288dfbbb74b /var/tmp btrfs subvol=@/var/tmp 0 0
UUID=6b7e374a-ae42-4f93-b6aa-d288dfbbb74b /.snapshots btrfs subvol=@/.snapshots 0 0

```

Comprendre le fichier /etc/fstab

Chaque ligne dans ce fichier contient 6 champs :

Champ 1	Champ 2	Champ 3	Champ 4	Champ 5	Champ 6
Fichier de bloc spécial ou UUID ou système de fichiers virtuel	Point de montage	Type de système de fichiers	Options séparées par des virgules	Utilisé par <i>dump</i> (1 = à dumper, 0 ou vide = à ignorer)	L'ordre de vérification par <i>fsck</i> des systèmes de fichiers au moment du démarrage

L'**UUID** (*Universally Unique Identifier*) est une chaîne d'une longueur de 128 bits. Les UUID sont créés automatiquement et d'une manière aléatoire lors de la création du filesystem sur la partition. Ils peuvent être modifiés par l'administrateur.

Options de Montage

Les options de montage les plus importants sont :

Option	Systèmes de Fichier	Description	Valeur par Défaut
defaults	Tous	Egal à rw, suid, dev, exec, auto, nouser, async	S/O
auto/noauto	Tous	Montage automatique/pas de montage automatique lors de l'utilisation de la commande mount -a	auto

Option	Systèmes de Fichier	Description	Valeur par Défaut
rw/ro	Tous	Montage en lecture-écriture/lecture seule	rw
suid/nosuid	Tous	Les bits SUID et SGID sont/ne sont pas pris en compte	suid
dev/nodev	Tous	Interprète/n'interprète pas les fichiers spéciaux de périphériques	dev
exec/noexec	Tous	Autorise/n'autorise pas l'exécution des programmes	exec
sync/async	Tous	Montage synchrone/asynchrone	async
user/nouser	Tous	Autorise/n'autorise pas un utilisateur à monter/démonter le système de fichier. Le point de montage est celui spécifié dans le fichier /etc/fstab. Seul l'utilisateur qui a monté le système de fichiers peut le démonter	S/O
users	Tous	Autorise tous les utilisateurs à monter/démonter le système de fichier	S/O
owner	Tous	Autorise le propriétaire du périphérique de le monter	S/O
atime/noatime	Norme POSIX	Inscrit/n'inscrit pas la date d'accès	atime
uid=valeur	Formats non-Linux	Spécifie le n° du propriétaire des fichiers pour les systèmes de fichiers non-Linux	root
gid=valeur	Formats non-Linux	Spécifie le n° du groupe propriétaire	S/O
umask=valeur	Formats non-Linux	Spécifie les permissions (droits d'accès/lecture/écriture)	S/O
dmask=valeur	Formats non-Linux	Spécifie les droits d'usage des dossiers (Obsolète, préférer dir_mode)	umask actuel
fmask=valeur	Formats non-Linux	Spécifie les droits d'usage des fichiers (Obsolète, préférer file_mode)	umask actuel

La Commande umount

Pour démonter un système de fichiers, on utilise la commande umount :

```
# umount /mnt/<répertoire_cible>
```

Options de la commande

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

```
SLES12SP1:~ # umount --help
```

Usage:

```
umount [-hV]
umount -a [options]
umount [options] <source> | <directory>
```

Options:

```
-a, --all                unmount all filesystems
-A, --all-targets        unmount all mountpoints for the given device in the
                        current namespace
-c, --no-canonicalize    don't canonicalize paths
-d, --detach-loop        if mounted loop device, also free this loop device
    --fake                dry run; skip the umount(2) syscall
-f, --force              force unmount (in case of an unreachable NFS system)
-i, --internal-only      don't call the umount.<type> helpers
-n, --no-mtab            don't write to /etc/mtab
-l, --lazy               detach the filesystem now, clean up things later
-O, --test-opts <list>   limit the set of filesystems (use with -a)
-R, --recursive          recursively unmount a target with all its children
-r, --read-only          in case unmounting fails, try to remount read-only
-t, --types <list>       limit the set of filesystem types
-v, --verbose            say what is being done

-h, --help               display this help and exit
-V, --version            output version information and exit
```

For more details see `umount(8)`.

Système de Fichiers Unix

Chaque partition sous un système Unix peut héberger les structures suivantes :

- superbloc

- inode
- bloc de données

Superbloc

Le superbloc contient :

- la taille des blocs
- la taille du système de fichiers
- le nombre de montages effectués pour ce système de fichiers
- un pointeur vers la racine du système de fichiers
- les pointeurs vers la liste des inodes libres
- les pointeurs vers la liste des blocs de données libres

Pour visualiser l'emplacement du Superbloc primaire et ses sauvegardes sur un système de fichiers de type **ext**, utilisez la commande suivante :

```
SLES11SP1:~ # mount | grep ext
/dev/sda2 on / type ext3 (rw,acl,user_xattr)
/dev/sda1 on /boot type ext3 (rw,acl,user_xattr)
SLES11SP1:~ # dumpe2fs /dev/sda1 | grep -i superblock
dumpe2fs 1.41.9 (22-Aug-2009)
  Primary superblock at 1, Group descriptors at 2-2
  Backup superblock at 8193, Group descriptors at 8194-8194
  Backup superblock at 24577, Group descriptors at 24578-24578
  Backup superblock at 40961, Group descriptors at 40962-40962
  Backup superblock at 57345, Group descriptors at 57346-57346
  Backup superblock at 73729, Group descriptors at 73730-73730
```

Pour réparer un système de fichiers extX en restaurant un Superbloc, utilisez la commande suivante :

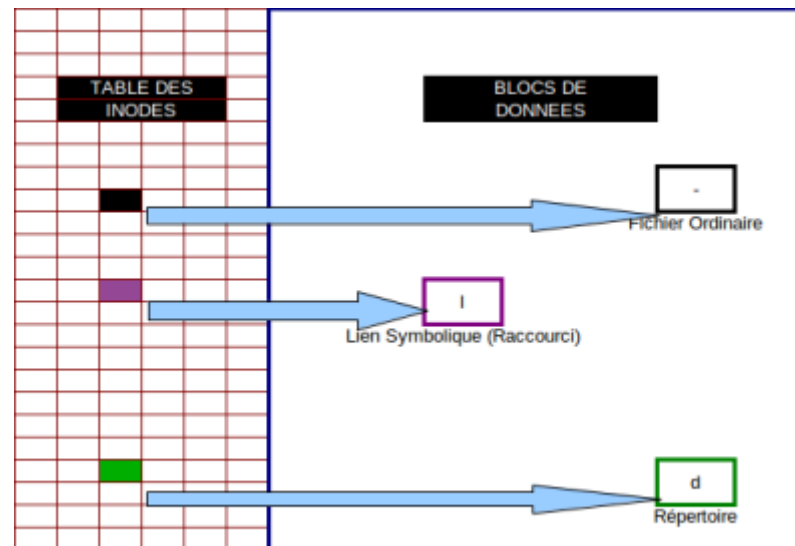
```
# e2fsck -f -b 8193 /dev/sda1 [Enter]
```

Inodes

Chaque fichier est représenté par un **inode**. L'inode contient :

- le type de fichier, soit -, **d**, **l**, **b**, **c**, **p**, **s**
- les droits d'accès, par exemple **rw** **x** **rw**- **r**-
- le nombre de liens physiques soit le nombre de noms
- l'UID du créateur ou l'UID affecté par la commande **chown** s'il y a eu une modification
- le GID du processus créateur ou le GID affecté par la commande **chgrp**
- la taille du fichier en octets
- la date de dernière modification de l'inode, soit le **ctime**
- la date de dernière modification du fichier, soit le **mtime**
- la date du dernier accès, soit le **atime**
- les adresses qui pointent vers les blocs de données du fichier

Graphiquement, on peut schématiser cette organisation de la façon suivante :



Pour mieux comprendre, tapez la commande suivante :

```
SLES12SP1:~ # ls -ld /dev/console /dev/sda1 /etc /etc/passwd
crw----- 1 root root 5, 1 Sep 30 10:23 /dev/console
brw-rw---- 1 root disk 8, 1 Sep 30 10:23 /dev/sda1
drwxr-xr-x 1 root root 5200 Sep 30 19:50 /etc
-rw-r--r-- 1 root root 1636 May  1 15:58 /etc/passwd
```

Le premier caractère de chaque ligne peut être un des suivants :

- - - un fichier
- **d** - un répertoire
- **l** - un lien symbolique
- **b** - un périphérique du type bloc
- **c** - un périphérique du type caractère
- **p** - un tube nommé pour la communication entre processus
- **s** - un socket dans un contexte réseau

Pour visualiser le numéro d'inode, utilisez l'option **-li** :

```
SLES12SP1:~ # ls -ldi /dev/console /dev/sda1 /etc /etc/passwd
4306 crw----- 1 root root 5, 1 Sep 28 10:37 /dev/console
6871 brw-rw---- 1 root disk 8, 1 Sep 28 10:37 /dev/sda1
 257 drwxr-xr-x 1 root root 4746 Sep 28 12:02 /etc
58930 -rw-r--r-- 1 root root 1335 Sep 20 13:34 /etc/passwd
```

Blocs de données

Les données sont stockées dans des blocs de données. Dans le cas d'un répertoire, le bloc de données contient une table qui référence les inodes et les noms des fichiers dans le répertoire. Cette table s'appelle une **table catalogue**.

Le nom d'un fichier n'est pas stocké dans l'inode mais dans une **table catalogue**. Cette particularité nous permet de donner deux noms différents au même fichier. Pour ajouter un nouveau nom à un fichier, il convient de créer un **lien physique**.

Liens Physiques

Un lien physique se crée en utilisant la commande suivante :

- In nom_du_fichier nom_supplémentaire

Pour illustrer ce point, tapez la ligne de commande suivante :

```
SLES12SP1:~ # cd /tmp; mkdir inode; cd inode; touch file1; ls -ali
total 0
442 drwxr-xr-x 1 root root 10 Sep 28 12:23 .
256 drwxrwxrwt 1 root root 112 Sep 28 12:23 ..
443 -rw-r--r-- 1 root root 0 Sep 28 12:23 file1
```

file1 indique un numéro d'inode **443** and a single name, indicated by the number **1** in the third column:

```
443 -rw-r--r-- 1 root root 0 Sep 28 12:23 file1
```

Créez maintenant un lien physique et visualisez le résultat de la commande :

```
SLES12SP1:/tmp/inode # ln file1 file2
SLES12SP1:/tmp/inode # ls -ali
total 0
442 drwxr-xr-x 1 root root 20 Sep 28 12:24 .
256 drwxrwxrwt 1 root root 112 Sep 28 12:23 ..
443 -rw-r--r-- 2 root root 0 Sep 28 12:23 file1
443 -rw-r--r-- 2 root root 0 Sep 28 12:23 file2
```

On peut maintenant voir deux lines, une pour file1 et la deuxième pour file2 :

```
443 -rw-r--r-- 2 root root 0 Sep 28 12:23 file1
443 -rw-r--r-- 2 root root 0 Sep 28 12:23 file2
```

Les deux fichiers, file1 et file2, sont référencés par le même inode. Le nombre de liens est donc augmenté de 1 (le numéro dans le troisième champs).



Important : Un lien physique ne peut être créé que dans le cas où les deux fichiers se trouvent dans le même filesystem et que le fichier source existe.

Liens Symboliques

Un lien symbolique est un **raccourci** vers un autre fichier ou répertoire. Un lien symbolique se crée en utilisant la commande suivante :

- `ln -s nom_du_fichier nom_raccourci`

Pour illustrer ce point, tapez la ligne de commande suivante :

```
# ln -s file1 file3 [Entrée]
```

Vous obtiendrez un résultat similaire à celui-ci :

```
SLES12SP1:/tmp/inode # ln -s file1 file3
SLES12SP1:/tmp/inode # ls -ali
total 4
442 drwxr-xr-x 1 root root 30 Sep 28 12:26 .
256 drwxrwxrwt 1 root root 112 Sep 28 12:23 ..
443 -rw-r--r-- 2 root root 0 Sep 28 12:23 file1
443 -rw-r--r-- 2 root root 0 Sep 28 12:23 file2
444 lrwxrwxrwx 1 root root 5 Sep 28 12:26 file3 -> file1
```

Notez que le lien symbolique est référencé par un autre inode. Le lien symbolique pointe vers file1.





Important : Un lien symbolique peut être créé même dans le cas où les deux fichiers se trouvent dans deux filesystems différents et même dans le cas où le fichier source n'existe pas.

<html>

Copyright © 2019 Hugh Norris.

</html>

From:

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

Permanent link:

<https://ittraining.team/doku.php?id=elearning:workbooks:opensuse:11:utilisateur:l101>

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

