

Version : **2022.01**

Dernière mise-à-jour : 2023/02/14 08:21

# LCF101 - Système de Fichiers

## Contenu du Module

- **LCF101 - Système de Fichiers**

- Contenu du Module
- Linux File Hierarchy System
  - RHEL/CentOS 6
  - RHEL/CentOS 7
- Types de Fichiers
- La Commande mount
  - Options de la commande
- Le Fichier /etc/fstab
  - Comprendre le fichier /etc/fstab
    - Options de Montage
- La Commande umount
  - Options de la commande
- Système de Fichiers Unix
  - Superbloc
  - Inodes
  - Blocs de données
  - Liens Physiques
  - Liens Symboliques

# 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**.

## RHEL/CentOS 6

```
[trainee@centos6 /]$ ls -l
total 98
dr-xr-xr-x.  2 root root  4096  9 août  12:52 bin
dr-xr-xr-x.  5 root root  1024  7 déc.  2014 boot
drwxr-xr-x. 19 root root  3820 25 août  11:29 dev
drwxr-xr-x. 119 root root 12288 25 août  11:28 etc
drwxr-xr-x.  3 root root  4096  3 mai   2013 home
dr-xr-xr-x.  20 root root 12288  9 août  12:52 lib
drwx-----.  2 root root 16384  3 mai   2013 lost+found
drwxr-xr-x.  2 root root  4096  7 déc.  2014 media
drwxr-xr-x.  2 root root     0 25 août  11:28 misc
drwxr-xr-x.  3 root root  4096  5 juil. 12:22 mnt
drwxr-xr-x.  2 root root     0 25 août  11:28 net
drwxr-xr-x.  6 root root  4096  7 déc.  2014 opt
dr-xr-xr-x. 154 root root     0 25 août  11:27 proc
dr-xr-x---. 10 root root  4096  9 août  12:58 root
dr-xr-xr-x.  2 root root 12288  9 août  12:52 sbin
drwxr-xr-x.  7 root root     0 25 août  11:27 selinux
drwxr-xr-x.  2 root root  4096 23 sept. 2011 srv
drwxr-xr-x. 13 root root     0 25 août  11:27 sys
drwxrwxrwt. 16 root root  4096 25 août  11:30 tmp
drwxr-xr-x. 13 root root  4096  3 mai   2013 usr
```

```
drwxr-xr-x. 22 root root 4096 9 août 12:50 var
```

- **/bin** : est une abréviation de **binary** ou binaires. Il contient des programmes tels ls.
- **/boot** : contient les fichiers nécessaires au démarrage du système.
- **/cgroup** : utilisé par le nouveau système de *Control Groups*.
- **/dev** : contient les nœuds utilisés pour accéder à tout type de matériel tel /dev/fd0 pour le lecteur de disquette. C'est le binaire *udev* 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 *fsck*.
- **/media** : contient des répertoires pour chaque système de fichiers monté ( accessible au système linux ) tels floppy, cdrom etc.
- **/misc** : contient des points de montage pour chaque répertoire local monté par l'automounter.
- **/mnt** : contient des répertoires pour chaque système de fichiers monté temporairement par root.
- **/net** : contient des points de montage pour chaque répertoire réseau monté par l'automounter.
- **/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
- **/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** : stocke 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.
- **/var** : contient des fichiers de taille variable.

## RHEL/CentOS 7

```
[trainee@centos7 /]$ ls -l
```

```
total 32
lrwxrwxrwx.  1 root root    7 Mar  8 13:41 bin -> usr/bin
dr-xr-xr-x.  4 root root 4096 Jun  4 15:00 boot
drwxr-xr-x. 19 root root 3280 Jul  7 15:55 dev
drwxr-xr-x. 131 root root 8192 Jul 23 17:05 etc
drwxr-xr-x.  4 root root   47 Jul  5 14:11 home
lrwxrwxrwx.  1 root root    7 Mar  8 13:41 lib -> usr/lib
lrwxrwxrwx.  1 root root    9 Mar  8 13:41 lib64 -> usr/lib64
drwxr-xr-x.  2 root root    6 Jun 10 2014 media
drwxr-xr-x.  3 root root   18 Jul  5 13:57 mnt
drwxr-xr-x.  4 root root   47 Jun  4 09:36 opt
dr-xr-xr-x. 177 root root    0 Jul  7 15:53 proc
dr-xr-x---.  5 root root 4096 Aug 25 11:31 root
drwxr-xr-x. 35 root root 1100 Jul 23 15:40 run
lrwxrwxrwx.  1 root root    8 Mar  8 13:41 sbin -> usr/sbin
drwxr-xr-x.  2 root root    6 Jun 10 2014 srv
dr-xr-xr-x. 13 root root    0 Jul  7 15:53 sys
drwxrwxrwt. 25 root root 4096 Jul 23 15:40 tmp
drwxr-xr-x. 13 root root 4096 Mar  8 13:41 usr
drwxr-xr-x. 22 root root 4096 Jul  7 15:53 var
```

- **/bin** : est une abréviation de **binary** ou binaires. Il contient des programmes tels ls. Sous RHEL/CentOS 7 il s'agit d'un lien symbolique qui pointe vers /usr/bin.
- **/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 udev 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. Sous RHEL/CentOS 7 il s'agit d'un lien symbolique qui pointe vers /usr/lib.
- **/lib64** : contient les bibliothèques 64 bits communes utilisées par les programmes ainsi que les modules. Sous RHEL/CentOS 7 il s'agit d'un lien symbolique qui pointe vers /usr/lib64.
- **/lost+found** : contient des fragments de fichiers endommagés et retrouvés par la commande fsck.

- **/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. Sous RHEL/CentOS 7 /var/run est un lien symbolique qui pointe vers /run.
- **/sbin** : contient des binaires, donc programmes, pour l'administration du système local. Sous RHEL/CentOS 7 il s'agit d'un lien symbolique qui pointe vers /usr/sbin.
- **/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** : stocke 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.
- **/var** : contient des fichiers de taille variable.

## 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)

Le fichiers normaux sont des fichiers textes, des tableaux ou des exécutables.

La longueur du nom de fichier est limité 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/<rédertoire_cible>
```

ou **/dev/<fichier\_spécial>** est le périphérique à monter et **/mnt/<rédertoire\_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 :

```
[trainee@centos6 /]$ su -
Password: fenestros
[root@centos6 /]# mount
/dev/sda2 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw)
/dev/sda1 on /boot type ext3 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)

[trainee@centos7 /]$ su -
Password: fenestros
[root@centos7 ~]# mount
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=1449668k,nr_inodes=362417,mode=755)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,seclabel)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,seclabel,gid=5,mode=620,ptmxmode=000)
```

```
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,mode=755)
tmpfs on /sys/fs/cgroup type tmpfs (rw,nosuid,nodev,noexec,seclabel,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/net_cls type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls)
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)
configfs on /sys/kernel/config type configfs (rw,relatime)
/dev/sda2 on / type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
selinuxfs on /sys/fs/selinux type selinuxfs (rw,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs
(rw,relatime,fd=32,pgrp=1,timeout=300,minproto=5,maxproto=5,direct)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,seclabel)
mqueue on /dev/mqueue type mqueue (rw,relatime,seclabel)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
nfsd on /proc/fs/nfsd type nfsd (rw,relatime)
/dev/sda1 on /boot type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
fusectl on /sys/fs/fuse/connections type fusectl (rw,relatime)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
```



**Important :** Notez que le système de fichiers par défaut est différent selon la version de RHEL. Sous RHEL 6 c'est l'**ext4** et sous RHEL 7 c'est le **xfs**. La comparaison des systèmes de fichiers Linux sera abordée dans le module **LCF204 - Gestion des Disques, des Systèmes de Fichiers et du Swap** de la formation **LCF200 - CentOS 7 Linux (RHEL 7) - Technician..**

## Options de la commande

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

```
[trainee@centos ~]$ mount --help
Utilisation : mount -V : afficher la version
    mount -h : afficher cette aide
    mount : lister les systèmes de fichiers montés
    mount -l : idem, incluant les étiquettes de volumes
```

Cela pour la partie informative. Suit ce qui porte sur le montage.

La commande est « `mount [-t type-sys-fichier] quoi où` ».

Les détails se trouvant dans `/etc/fstab` peuvent être omis.

```
mount -a [-t|-0] ... : monter tout ce qui est listé dans /etc/fstab
mount périphérique : monter le périphérique à l'endroit connu
mount répertoire : monter le périphérique connu ici
mount -t type périph rép : commande de montage ordinaire
```

Noter que celle-ci ne monte pas réellement un périphérique, elle monte un système de fichiers (de type donné) trouvé sur le périphérique.

Elle peut aussi monter une arborescence de rép. déjà visible ailleurs :

```
mount --bind ancien-rép nouveau-rép
```

ou déplacer une sous-arborescence:

```
mount --move ancien-rép nouveau-rép
```

Elle peut changer le type de montage d'un rép. :

```
mount --make-shared rép.
```

```
mount --make-slave rép.
```

```
mount --make-private rép.
```

```
mount --make-unbindable rép.
```

Elle peut changer le type de tous les points de montage d'une sous-arborescence contenue dans le rép.:

```
mount --make-rshared rép.
```

```
mount --make-rslave rép.
```

```
mount --make-rprivate rép.
```

```
mount --make-runbindable rép.
```

Un périph. peut être nommé, comme `/dev/hda1` ou `/dev/cdrom`, ou repéré par l'étiquette, avec `-L` étiqu. ou par UUID, avec `-U` uuid . Autres options: `[-nfFrsvw] [-o options] [-p descr_fic_mots_passe]`. Pour plus de détails, tapez "man 8 mount".

## 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 :

```
[root@centos6 ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Fri May  3 13:33:42 2013
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=b9f29672-c84e-4d3b-b132-189758a084eb /          ext4    defaults      1 1
UUID=01baf03d-df0d-479b-b3e4-81ce63b8dec3 /boot       ext4    defaults      1 2
UUID=2646a33a-65f3-4501-9ced-9459435fd774 swap       swap    defaults      0 0
tmpfs          /dev/shm          tmpfs   defaults      0 0
devpts         /dev/pts          devpts  gid=5,mode=620 0 0
sysfs          /sys              sysfs   defaults      0 0
proc           /proc              proc    defaults      0 0
```

```
[root@centos7 ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Sun Mar  8 12:38:10 2015
```

```

#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=b35de665-5ec8-4226-a533-58a1b567ac91 /           xfs      defaults      1  1
UUID=e8d3bd48-1386-411c-9675-41c3f8f1a309 /boot        xfs      defaults      1  2
UUID=11a4d11d-81e4-46a7-82e0-7796cd597dc9 swap         swap      defaults      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, uid, dev, exec, auto, nouser, async	S/O
auto/noauto	Tous	Montage automatique/pas de montage automatique lors de l'utilisation de la commande <b>mount -a</b>	auto
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

Option	Systèmes de Fichier	Description	Valeur par Défaut
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
dir_mode=valeur	Formats non-Linux	Spécifie les droits d'usage des dossiers	umask actuel
fmask=valeur	Formats non-Linux	Spécifie les droits d'usage des fichiers (Obsolète, préférer file_mode)	umask actuel
file_mode=valeur	Formats non-Linux	Spécifie les droits d'usage des fichiers	umask actuel

## La Commande umount

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

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

ou

```
# umount /dev/cdrom
```

## Options de la commande

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

```
[trainee@centos ~]$ umount --help
Utilisation : umount -h | -V
    umount -a [-d] [-f] [-r] [-n] [-v] [-t typevfs] [-O opts]
    umount [-d] [-f] [-r] [-n] [-v] spécial | noeud...
```

## Système de Fichiers Unix

Chaque partition sous un système Unix peut héberger une des structures suivantes :

- superbloc
- inode
- bloc de données
- blocs d'indirection

### 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

Le Superbloc est dupliqué tous les 8 ou 16Mo sous ext3 et ext4. Pour réparer un système de fichiers en restaurant un Superbloc, utilisez la commande suivante :

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

Pour visualiser l'emplacement du Superbloc primaire et ses sauvegardes, utilisez la commande suivante :

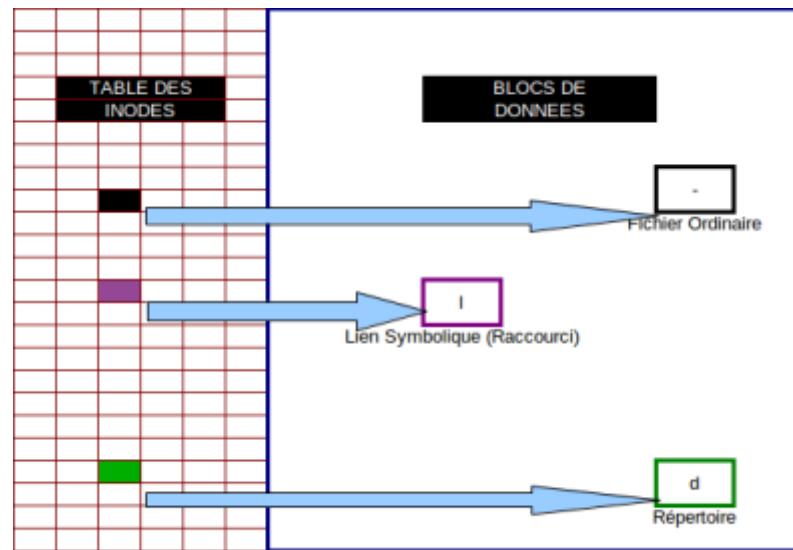
```
[root@centos6 ~]# dumpe2fs /dev/sda1 | grep -i superbloc
dumpe2fs 1.41.12 (17-May-2010)
superbloc Primaire à 1, Descripteurs de groupes à 2-2
superbloc Secours à 8193, Descripteurs de groupes à 8194-8194
superbloc Secours à 24577, Descripteurs de groupes à 24578-24578
superbloc Secours à 40961, Descripteurs de groupes à 40962-40962
superbloc Secours à 57345, Descripteurs de groupes à 57346-57346
superbloc Secours à 73729, Descripteurs de groupes à 73730-73730
```

## 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 **rwx 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 :

```
[root@centos6 ~]# ls -ld /dev/console /dev/ram0 /etc /etc/passwd
crw----- 1 root root 5, 1 20 oct. 15:52 /dev/console
brw-rw---- 1 root disk 1, 0 20 oct. 15:52 /dev/ram0
drwxr-xr-x 117 root root 12288 20 oct. 15:53 /etc
-rw-r--r-- 1 root root 1890 3 oct. 18:09 /etc/passwd
```

```
[root@centos7 ~]# ls -ld /dev/console /dev/ram0 /etc /etc/passwd
ls: cannot access /dev/ram0: No such file or directory
crw-----. 1 root root 5, 1 Jul 7 15:54 /dev/console
drwxr-xr-x. 131 root root 8192 Aug 25 14:27 /etc
-rw-r--r--. 1 root root 2103 Jul 5 14:11 /etc/passwd
```

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

- - - un fichier
- **d** - un répertoire

- **I** - 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 **-i** :

```
[root@centos6 ~]# ls -ldi /dev/console /dev/ram0 /etc /etc/passwd
5153 crw----- 1 root root 5, 1 20 oct. 15:52 /dev/console
7377 brw-rw---- 1 root disk 1, 0 20 oct. 15:52 /dev/ram0
 15 drwxr-xr-x 117 root root 12288 20 oct. 15:53 /etc
13564 -rw-r--r-- 1 root root 1890 3 oct. 18:09 /etc/passwd
```

```
root@centos7 ~]# ls -ldi /dev/console /dev/ram0 /etc /etc/passwd
ls: cannot access /dev/ram0: No such file or directory
 1043 crw----- 1 root root 5, 1 Jul 7 15:54 /dev/console
 8388737 drwxr-xr-x. 131 root root 8192 Aug 25 14:27 /etc
11114576 -rw-r--r--. 1 root root 2103 Jul 5 14:11 /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 :

- `ln nom_du_fichier nom_supplémentaire`

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

```
[root@centos7 ~]# cd /tmp; mkdir inode; cd inode; touch fichier1; ls -ali
total 4
27689296 drwxr-xr-x. 2 root root 21 Oct 15 15:23 .
25165953 drwxrwxrwt. 23 root root 4096 Oct 15 15:23 ..
27689297 -rw-r--r--. 1 root root 0 Oct 15 15:23 fichier1
```

Notez bien le numéro de l'inode du fichier **fichier1**. Notez aussi que le numéro dans le troisième champs de la ligne de fichier1 a la valeur **1** :

```
27689297 -rw-r--r--. 1 root root 0 Oct 15 15:23 fichier1
```

Créez maintenant un lien physique :

```
[root@centos7 inode]# ln fichier1 fichier2
[root@centos7 inode]# ls -ali
total 4
27689296 drwxr-xr-x. 2 root root 36 Oct 15 15:24 .
25165953 drwxrwxrwt. 23 root root 4096 Oct 15 15:23 ..
27689297 -rw-r--r--. 2 root root 0 Oct 15 15:23 fichier1
27689297 -rw-r--r--. 2 root root 0 Oct 15 15:23 fichier2
```

Notez les deux lignes suivantes :

```
27689297 -rw-r--r--. 2 root root 0 Oct 15 15:23 fichier1
```

```
27689297 -rw-r--r--. 2 root root 0 Oct 15 15:23 fichier2
```

Les deux fichiers, fichier1 et fichier2, 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 fichier1 fichier3 [Entrée]
```

Vous obtiendrez un résultat similaire à celui-ci :

```
[root@centos7 inode]# ln -s fichier1 fichier3
[root@centos7 inode]# ls -ali
total 4
27689296 drwxr-xr-x.  2 root root  51 Oct 15 15:25 .
25165953 drwxrwxrwt. 23 root root 4096 Oct 15 15:23 ..
27689297 -rw-r--r--.  2 root root  0 Oct 15 15:23 fichier1
27689297 -rw-r--r--.  2 root root  0 Oct 15 15:23 fichier2
27689298 lrwxrwxrwx.  1 root root  8 Oct 15 15:25 fichier3 -> fichier1
```

Notez que le lien symbolique est référencé par un autre inode. Le lien symbolique pointe vers le fichier1.



**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.

Copyright © 2023 Hugh Norris.

From:

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



Permanent link:

<https://www.ittraining.team/doku.php?id=elearning:workbooks:centos:6:utilisateur:l101>

Last update: **2023/02/14 08:21**