

Dernière mise-à-jour : 2020/07/30 10:02

# LDF101 - Système de Fichiers

## Contenu du Module

- **LDF101 - Système de Fichiers**
  - Contenu du Module
  - Linux File Hierarchy System
    - Debian 6
    - Debian 7
    - Debian 8
    - Debian 9
  - 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**.

## Debian 6

```
trainee@debian6:~$ cd /
trainee@debian6:/$ ls -l
total 92
drwxr-xr-x  2 root root  4096  6 déc.  2014 bin
drwxr-xr-x  3 root root  4096  6 déc.  2014 boot
drwxr-xr-x 15 root root 3120 25 août 16:51 dev
drwxr-xr-x 121 root root 12288  4 juil. 15:05 etc
drwxr-xr-x  3 root root  4096 24 avril 2011 home
lrwxrwxrwx  1 root root    28 24 avril 2011 initrd.img -> boot/initrd.img-2.6.32-5-686
drwxr-xr-x 12 root root 12288  6 déc.  2014 lib
drwx----- 2 root root 16384 24 avril 2011 lost+found
drwxr-xr-x  3 root root  4096 24 avril 2011 media
drwxr-xr-x  2 root root  4096 14 déc.  2010 mnt
drwxr-xr-x  4 root root  4096  4 déc.  2011 opt
dr-xr-xr-x 128 root root    0  4 juil. 14:34 proc
drwx----- 11 root root  4096  7 déc.  2014 root
drwxr-xr-x  2 root root  4096  6 déc.  2014 sbin
drwxr-xr-x  2 root root  4096 21 juil. 2010 selinux
drwxr-xr-x  2 root root  4096 24 avril 2011 srv
drwxr-xr-x 12 root root    0  4 juil. 14:34 sys
drwxrwxrwt 10 root root  4096 25 août 16:34 tmp
drwxr-xr-x 10 root root  4096 24 avril 2011 usr
drwxr-xr-x 14 root root  4096 24 avril 2011 var
```

```
lrwxrwxrwx  1 root root   25 24 avril  2011 vmlinuz -> boot/vmlinuz-2.6.32-5-686
```

## Debian 7

```
trainee@debian7:~$ cd /
trainee@debian7:/$ ls -l
total 84
drwxr-xr-x  2 root root  4096 juin  26 16:06 bin
drwxr-xr-x  3 root root  4096 juin  26 16:09 boot
drwxr-xr-x 14 root root  3160 juil.  4 16:36 dev
drwxr-xr-x 133 root root 12288 juil.  4 16:36 etc
drwxr-xr-x  3 root root  4096 juin  26 15:42 home
lrwxrwxrwx  1 root root    32 juin  26 15:35 initrd.img -> /boot/initrd.img-3.2.0-4-686-pae
lrwxrwxrwx  1 root root    28 juin  26 15:35 initrd.img.old -> /boot/initrd.img-3.2.0-4-486
drwxr-xr-x 15 root root  4096 juin  26 16:01 lib
drwx----- 2 root root 16384 juin  26 15:35 lost+found
drwxr-xr-x  3 root root  4096 juil. 15  2014 media
drwxr-xr-x  2 root root  4096 juil.  7  2014 mnt
drwxr-xr-x  2 root root  4096 juil. 15  2014 opt
dr-xr-xr-x 131 root root    0 juin  26 16:17 proc
drwx----- 4 root root  4096 juil.  4 15:14 root
drwxr-xr-x 20 root root   880 juil.  4 16:36 run
drwxr-xr-x  2 root root  4096 juin  26 16:07 sbin
drwxr-xr-x  2 root root  4096 juin 10  2012 selinux
drwxr-xr-x  2 root root  4096 juil. 15  2014 srv
drwxr-xr-x 13 root root    0 juin  26 16:17 sys
drwxrwxrwt  9 root root  4096 juil.  4 16:17 tmp
drwxr-xr-x 10 root root  4096 juin  26 15:39 usr
drwxr-xr-x 12 root root  4096 juin  26 15:39 var
lrwxrwxrwx  1 root root    28 juin  26 15:39 vmlinuz -> boot/vmlinuz-3.2.0-4-686-pae
lrwxrwxrwx  1 root root    24 juin  26 15:39 vmlinuz.old -> boot/vmlinuz-3.2.0-4-486
```

## Debian 8

```
trainee@debian8:~$ cd /
trainee@debian8:/$ ls -l
total 84
drwxr-xr-x  2 root root  4096 juin  28 16:31 bin
drwxr-xr-x  3 root root  4096 juin  28 16:31 boot
drwxr-xr-x 17 root root  3080 juil. 23 17:04 dev
drwxr-xr-x 125 root root 12288 août 27 12:18 etc
drwxr-xr-x  3 root root  4096 juin  28 16:30 home
lrwxrwxrwx  1 root root    33 juin  28 16:26 initrd.img -> /boot/initrd.img-3.16.0-4-686-pae
lrwxrwxrwx  1 root root    29 juin  28 16:26 initrd.img.old -> /boot/initrd.img-3.16.0-4-586
drwxr-xr-x 18 root root  4096 juin  28 16:31 lib
drwxr-xr-x  2 root root  4096 juin   6 17:07 live-build
drwx----- 2 root root 16384 juin  28 16:26 lost+found
drwxr-xr-x  3 root root  4096 juin   6 16:32 media
drwxr-xr-x  2 root root  4096 juin   6 16:32 mnt
drwxr-xr-x  3 root root  4096 juin  28 16:38 opt
dr-xr-xr-x 138 root root    0 juin  28 16:41 proc
drwx----- 2 root root  4096 juin  28 16:35 root
drwxr-xr-x 24 root root   820 août 27 12:18 run
drwxr-xr-x  2 root root  4096 juin  28 16:39 sbin
drwxr-xr-x  2 root root  4096 juin   6 16:32 srv
dr-xr-xr-x 13 root root    0 juin  28 16:41 sys
drwxrwxrwt 12 root root  4096 juil. 23 16:17 tmp
drwxr-xr-x 10 root root  4096 juin  28 16:29 usr
drwxr-xr-x 11 root root  4096 juin  28 16:29 var
lrwxrwxrwx  1 root root    29 juin  28 16:29 vmlinuz -> boot/vmlinuz-3.16.0-4-686-pae
lrwxrwxrwx  1 root root    25 juin  28 16:29 vmlinuz.old -> boot/vmlinuz-3.16.0-4-586
```

## Debian 9

```
trainee@debian9:~$ cd /
trainee@debian9:/$ ls -l
total 76
drwxr-xr-x  2 root root 4096 janv. 22 13:36 bin
drwxr-xr-x  3 root root 4096 janv. 22 13:38 boot
drwxr-xr-x 16 root root 2980 mars 19 11:59 dev
drwxr-xr-x 109 root root 4096 mars 19 11:59 etc
drwxr-xr-x  3 root root 4096 janv. 22 13:39 home
lrwxrwxrwx  1 root root   29 janv. 22 13:18 initrd.img -> boot/initrd.img-4.9.0-8-amd64
lrwxrwxrwx  1 root root   29 janv. 22 13:18 initrd.img.old -> boot/initrd.img-4.9.0-8-amd64
drwxr-xr-x 15 root root 4096 janv. 22 13:37 lib
drwxr-xr-x  2 root root 4096 janv. 22 13:17 lib64
drwx----- 2 root root 16384 janv. 22 13:17 lost+found
drwxr-xr-x  3 root root 4096 janv. 22 13:17 media
drwxr-xr-x  2 root root 4096 janv. 22 13:17 mnt
drwxr-xr-x  2 root root 4096 janv. 22 13:17 opt
dr-xr-xr-x 97 root root    0 mars 19 11:58 proc
drwx----- 3 root root 4096 janv. 22 13:40 root
drwxr-xr-x 19 root root  600 mars 19 12:02 run
drwxr-xr-x  2 root root 4096 janv. 22 13:39 sbin
drwxr-xr-x  2 root root 4096 janv. 22 13:17 srv
dr-xr-xr-x 13 root root    0 mars 19 11:58 sys
drwxrwxrwt 10 root root 4096 mars 19 11:59 tmp
drwxr-xr-x 10 root root 4096 janv. 22 13:17 usr
drwxr-xr-x 11 root root 4096 janv. 22 13:17 var
lrwxrwxrwx  1 root root   26 janv. 22 13:18 vmlinuz -> boot/vmlinuz-4.9.0-8-amd64
lrwxrwxrwx  1 root root   26 janv. 22 13:18 vmlinuz.old -> boot/vmlinuz-4.9.0-8-amd64
```

- **/bin** : est 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 *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 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.
- **/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 Debian 7 et 8, /var/run est un lien symbolique qui pointe 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** : 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/<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.



**A faire** : Connectez-vous à votre machine virtuelle en tant que **trainee** avec le mot de passe **trainee**. Ouvrez un terminal via les menus **Applications > Favorites > Terminal**. Tapez la commande **su -** et appuyez sur la touche **↵ Entrée**. Indiquez le mot de passe **fenestros**. Vous êtes maintenant connecté en tant que l'administrateur **root** et vous pouvez reproduire les exemples qui suivent.

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@debian6:/$ cd ~
trainee@debian6:~$ su -
Mot de passe : fenestros
trainee@debian6:/$ cd ~
root@debian6:~# mount
/dev/sda1 on / type ext3 (rw,errors=remount-ro)
tmpfs on /lib/init/rw type tmpfs (rw,nosuid,mode=0755)
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
udev on /dev type tmpfs (rw,mode=0755)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
```

```
devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=620)
fusectl on /sys/fs/fuse/connections type fusectl (rw)
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc (rw,noexec,nosuid,nodev)
```

```
trainee@debian7:/$ cd ~
trainee@debian7:~$ su -
Mot de passe : fenestros
root@debian7:~# mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,relatime,size=10240k,nr_inodes=127976,mode=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=103420k,mode=755)
/dev/disk/by-uuid/0f1cdb0c-8bd7-45ec-8d99-064292047bdb on / type ext4 (rw,relatime,errors=remount-
ro,user_xattr,barrier=1,data=ordered)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs on /run/shm type tmpfs (rw,nosuid,nodev,noexec,relatime,size=597180k)
rpc_pipefs on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc (rw,nosuid,nodev,noexec,relatime)
```

```
trainee@debian8:/$ cd ~
trainee@debian8:~$ su -
Mot de passe : fenestros
root@debian8:~# mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,relatime,size=10240k,nr_inodes=126281,mode=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,relatime,size=206376k,mode=755)
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro,data=ordered)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
```

```
cgroup on /sys/fs/cgroup/systemd type cgroup
(rw,nosuid,nodev,noexec,relatime,xattr,release_agent=/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,cpu,cpuacct)
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,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls,net_prio)
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)
tmpfs on /etc/machine-id type tmpfs (ro,relatime,size=206376k,mode=755)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs
(rw,relatime,fd=22,prgrp=1,timeout=300,minproto=5,maxproto=5,direct)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime)
mqueue on /dev/mqueue type mqueue (rw,relatime)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
fusectl on /sys/fs/fuse/connections type fusectl (rw,relatime)
rpc_pipefs on /run/rpc_pipefs type rpc_pipefs (rw,relatime)
tmpfs on /run/user/119 type tmpfs (rw,nosuid,nodev,relatime,size=103188k,mode=700,uid=119,gid=127)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,size=103188k,mode=700,uid=1000,gid=1000)
```

```
trainee@debian9:/$ cd ~
trainee@debian9:~$ su -
Mot de passe : fenestros
root@debian9:~# mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relatime,size=1014992k,nr_inodes=253748,mode=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=205256k,mode=755)
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro,data=ordered)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
```

```
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
cgroup on /sys/fs/cgroup/systemd type cgroup
(rw,nosuid,nodev,noexec,relatime,xattr,release_agent=/lib/systemd/systemd-cgroups-agent,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls,net_prio)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
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/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs
(rw,relatime,fd=35,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=9229)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime)
mqueue on /dev/mqueue type mqueue (rw,relatime)
tmpfs on /run/user/112 type tmpfs (rw,nosuid,nodev,relatime,size=205252k,mode=700,uid=112,gid=115)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,size=205252k,mode=700,uid=1000,gid=1000)
```



**Important** : Notez que le système de fichiers par défaut est différent selon la version de Debian. Sous Debian 6 c'est l'**ext3**, sous Debian 7 et 8 c'est l'**ext4**. La comparaison des systèmes de fichiers Linux sera abordée dans le cours **Gestion des Disques, des Systèmes de Fichiers et du Swap**.

## Options de la commande

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

```
root@debian9:~# mount --help
```

**Utilisation :**

```
mount [-lhV]
mount -a [options]
mount [options] [--source] <source> | [--target] <répertoire>
mount [options] <source> <répertoire>
mount <opération> <pointdemontage> [<cible>]
```

Monter un système de fichiers.

**Options :**

-a, --all	monter tous systèmes de fichiers indiqués dans fstab
-c, --no-canonicalize	ne pas canoniser les chemins
-f, --fake	répétition, sauter l'appel système mount(2)
-F, --fork	créer un processus fils pour chaque périphérique (utiliser avec -a)
-T, --fstab <chemin>	autre fichier pour /etc/fstab
-i, --internal-only	don't call the mount.<type> helpers
-l, --show-labels	show also filesystem labels
-n, --no-mtab	ne pas écrire dans /etc/mtab
-o, --options <liste>	liste séparée par des virgules d'options de montage
-O, --test-opts <liste>	limiter l'ensemble de systèmes de fichiers (utiliser avec -a)
-r, --read-only	monter le système de fichiers en lecture seule (identique à -o ro)
-t, --types <liste>	limiter l'ensemble de types de système de fichiers
--source <source>	indiquer explicitement la source (chemin, étiqu., UUID)
--target <cible>	indiquer explicitement le point de montage
-v, --verbose	expliquer les actions en cours
-w, --rw, --read-write	mount the filesystem read-write (default)
-h, --help	afficher cette aide et quitter
-V, --version	afficher les informations de version et quitter

**Source :**

```
-L, --label <étiquette>  synonyme de LABEL=<étiquette>
-U, --uuid <uuid>        synonyme de UUID=<uuid>
LABEL=<étiquette>         indiquer périph. par étiquette de système de fichiers
UUID=<uuid>               indiquer périphérique par UUID de système de fichiers
PARTLABEL=<étiquette>     indiquer le périphérique par étiquette de partition
PARTUUID=<uuid>           indiquer le périphérique par UUID de partition
<périphérique>           indiquer le périphérique par chemin
<répertoire>             pt de montage pour remontages (consultez --bind/rbind)
<fichier>               fichier régulier pour configuration de périph. boucle
```

#### Opérations :

```
-B, --bind                monter un sous-répertoire ailleurs (comme -o bind)
-M, --move                déplacer un sous-répertoire ailleurs
-R, --rbind               monter un sous-rép. et tous les sous-montages ailleurs
--make-shared             marquer un sous-répertoire comme partagé
--make-slave              marquer un sous-répertoire comme esclave
--make-private            marquer un sous-répertoire comme privé
--make-unbindable         marquer un sous-répertoire comme non remontable
--make-rshared            marquer récursivement tout un sous-répertoire partagé
--make-rslave             marquer récursivement tout un sous-répertoire esclave
--make-rprivate           marquer récursivement tout un sous-répertoire privé
--make-runbindable        marquer récursivement tout un sous-rép. non remontable
```

Consultez `mount(8)` pour obtenir des précisions complémentaires.

## 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@debian6:~# cat /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>
proc                /proc                proc    defaults           0        0
# / was on /dev/sda1 during installation
UUID=a42a1ddd-14bc-4dde-a537-e6c1b984a782 /                    ext3     errors=remount-ro 0        1
# swap was on /dev/sda5 during installation
UUID=e21d8931-21ca-4ab3-9fbb-bd71657b312e none                 swap     sw                0        0
```

```
root@debian7:~# cat /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=0f1cdb0c-8bd7-45ec-8d99-064292047bdb /                    ext4     errors=remount-ro 0        1
# swap was on /dev/sda5 during installation
UUID=ae5f67d1-6fad-487e-ad41-d53992a75755 none                 swap     sw                0        0
```

```
root@debian8:~# cat /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=0ac29bda-b6bb-41c2-b47a-03fecb95bc87 /          ext4    errors=remount-ro 0      1
# swap was on /dev/sda5 during installation
UUID=fa3153a5-5ef4-46fe-9115-d3773a06d283 none        swap      sw              0      0

```

```

root@debian9:~# cat /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=b29b93ed-8de4-4980-a407-15bba3ad09ba /          ext4    errors=remount-ro 0      1
# swap was on /dev/sda5 during installation
UUID=479bcf18-634c-42af-bb34-d9b7c97230d1 none        swap      sw              0      0
/dev/sr0      /media/cdrom0    udf,iso9660 user,noauto     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 <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
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>
```

## Options de la commande

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

```
root@debian9:~# umount --help
```

Utilisation :

```
umount [-hV]
```

```
umount -a [options]
```

```
umount [options] <source> | <répertoire>
```

Démonter des systèmes de fichiers.

Options :

-a, --all	démonter tous les systèmes de fichiers
-A, --all-targets	démonter tous les points de montage pour le périphérique dans l'espace de noms actuel
-c, --no-canonicalize	ne pas canoniser les chemins
-d, --detach-loop	libérer aussi le périphérique boucle s'il est monté
--fake	répétition, sauter l'appel système umount(2)
-f, --force	forcer le démontage (pour un montage NFS inaccessible)
-i, --internal-only	ne pas appeler les auxiliaires umount.<type>
-n, --no-mtab	ne pas écrire dans /etc/mtab
-l, --lazy	détacher le système de fichiers maintenant, et nettoyer les choses plus tard
-O, --test-opts <liste>	limiter l'ensemble de systèmes de fichiers (utiliser avec -a)
-R, --recursive	démonter récursivement une cible avec tous ces enfants
-r, --read-only	essayer de remonter en lecture seule en cas d'échec du démontage

```
-t, --types <liste>    limiter l'ensemble de systèmes de fichiers par types  
-v, --verbose          expliquer les actions en cours
```

```
-h, --help            afficher cette aide et quitter  
-V, --version         afficher les informations de version et quitter
```

Consultez umount(8) pour obtenir des précisions complémentaires.

## 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@debian6:~# dumpe2fs /dev/sda1 | grep -i superbloc
dumpe2fs 1.41.12 (17-May-2010)
superbloc Primaire à 0, Descripteurs de groupes à 1-1
superbloc Secours à 32768, Descripteurs de groupes à 32769-32769
superbloc Secours à 98304, Descripteurs de groupes à 98305-98305
superbloc Secours à 163840, Descripteurs de groupes à 163841-163841
superbloc Secours à 229376, Descripteurs de groupes à 229377-229377
superbloc Secours à 294912, Descripteurs de groupes à 294913-294913
superbloc Secours à 819200, Descripteurs de groupes à 819201-819201
superbloc Secours à 884736, Descripteurs de groupes à 884737-884737
```

```
root@debian7:~# dumpe2fs /dev/sda1 | grep -i superbloc
dumpe2fs 1.42.5 (29-Jul-2012)
superbloc Primaire à 0, Descripteurs de groupes à 1-1
superbloc Secours à 32768, Descripteurs de groupes à 32769-32769
superbloc Secours à 98304, Descripteurs de groupes à 98305-98305
superbloc Secours à 163840, Descripteurs de groupes à 163841-163841
superbloc Secours à 229376, Descripteurs de groupes à 229377-229377
superbloc Secours à 294912, Descripteurs de groupes à 294913-294913
superbloc Secours à 819200, Descripteurs de groupes à 819201-819201
superbloc Secours à 884736, Descripteurs de groupes à 884737-884737
superbloc Secours à 1605632, Descripteurs de groupes à 1605633-1605633
```

```
root@debian8:~# dumpe2fs /dev/sda1 | grep -i superbloc
dumpe2fs 1.42.12 (29-Aug-2014)
superbloc Primaire à 0, Descripteurs de groupes à 1-1
superbloc Secours à 32768, Descripteurs de groupes à 32769-32769
superbloc Secours à 98304, Descripteurs de groupes à 98305-98305
superbloc Secours à 163840, Descripteurs de groupes à 163841-163841
superbloc Secours à 229376, Descripteurs de groupes à 229377-229377
```

```
superbloc Secours à 294912, Descripteurs de groupes à 294913-294913
superbloc Secours à 819200, Descripteurs de groupes à 819201-819201
superbloc Secours à 884736, Descripteurs de groupes à 884737-884737
superbloc Secours à 1605632, Descripteurs de groupes à 1605633-1605633
```

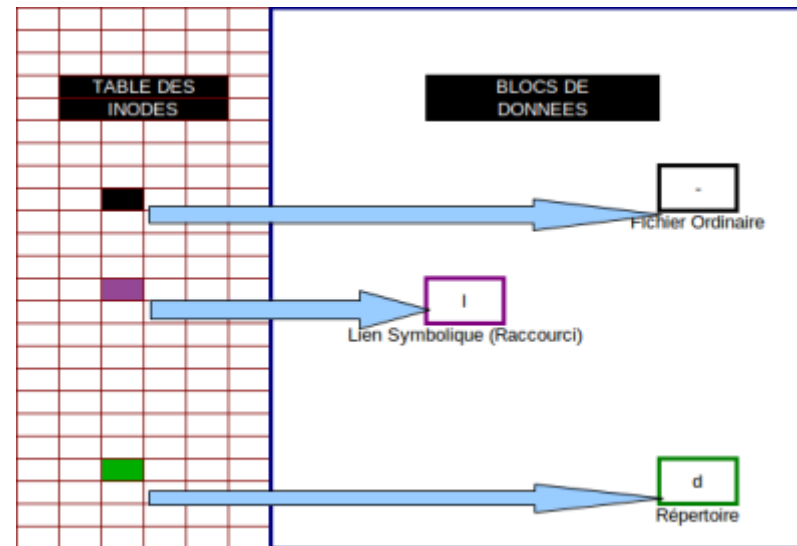
```
root@debian9:~# dumpe2fs /dev/sda1 | grep -i superbloc
dumpe2fs 1.43.4 (31-Jan-2017)
superbloc Primaire à 0, Descripteurs de groupes à 1-3
superbloc Secours à 32768, Descripteurs de groupes à 32769-32771
superbloc Secours à 98304, Descripteurs de groupes à 98305-98307
superbloc Secours à 163840, Descripteurs de groupes à 163841-163843
superbloc Secours à 229376, Descripteurs de groupes à 229377-229379
superbloc Secours à 294912, Descripteurs de groupes à 294913-294915
superbloc Secours à 819200, Descripteurs de groupes à 819201-819203
superbloc Secours à 884736, Descripteurs de groupes à 884737-884739
superbloc Secours à 1605632, Descripteurs de groupes à 1605633-1605635
superbloc Secours à 2654208, Descripteurs de groupes à 2654209-2654211
superbloc Secours à 4096000, Descripteurs de groupes à 4096001-4096003
```

## 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 :

```
root@debian6:~# ls -ld /dev/console /dev/loop0 /etc /etc/passwd
crw----- 1 root root 5, 1 4 juil. 14:34 /dev/console
brw-rw---- 1 root disk 7, 0 4 juil. 14:34 /dev/loop0
drwxr-xr-x 121 root root 12288 4 juil. 15:05 /etc
-rw-r--r-- 1 root root 1298 27 avril 2011 /etc/passwd
```

```
root@debian7:~# ls -ld /dev/console /dev/loop0 /etc /etc/passwd
crw----- 1 root root 5, 1 juin 26 16:17 /dev/console
brw-rw---T 1 root disk 7, 0 juin 26 16:17 /dev/loop0
drwxr-xr-x 133 root root 12288 juil. 4 16:36 /etc
-rw-r--r-- 1 root root 1657 juin 26 15:42 /etc/passwd
```

```
root@debian8:~# ls -ld /dev/console /dev/sda /etc /etc/passwd
crw----- 1 root root 5, 1 Oct 15 15:27 /dev/console
```

```
brw-rw---- 1 root disk 8, 0 Oct 15 15:27 /dev/sda
drwxr-xr-x 134 root root 12288 Oct 15 15:27 /etc
-rw-r--r-- 1 root root 2202 Oct 15 13:27 /etc/passwd
```

```
root@debian9:~# ls -ld /dev/console /dev/sda /etc /etc/passwd
crw----- 1 root root 5, 1 mars 19 11:58 /dev/console
brw-rw---- 1 root disk 8, 0 mars 19 11:58 /dev/sda
drwxr-xr-x 109 root root 4096 mars 19 11:59 /etc
-rw-r--r-- 1 root root 1962 janv. 22 13:39 /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** :

```
root@debian6:~# ls -ldi /dev/console /dev/loop0 /etc /etc/passwd
1251 crw----- 1 root root 5, 1 4 juil. 14:34 /dev/console
2041 brw-rw---- 1 root disk 7, 0 4 juil. 14:34 /dev/loop0
194689 drwxr-xr-x 121 root root 12288 4 juil. 15:05 /etc
201193 -rw-r--r-- 1 root root 1298 27 avril 2011 /etc/passwd
```

```
root@debian7:~# ls -ldi /dev/console /dev/loop0 /etc /etc/passwd
1196 crw----- 1 root root 5, 1 juin 26 16:17 /dev/console
3269 brw-rw---T 1 root disk 7, 0 juin 26 16:17 /dev/loop0
260609 drwxr-xr-x 133 root root 12288 juil. 4 16:36 /etc
262280 -rw-r--r-- 1 root root 1657 juin 26 15:42 /etc/passwd
```

```
root@debian8:~# ls -ldi /dev/console /dev/sda /etc /etc/passwd
5676 crw----- 1 root root 5, 1 Oct 15 15:27 /dev/console
6938 brw-rw---- 1 root disk 8, 0 Oct 15 15:27 /dev/sda
260609 drwxr-xr-x 134 root root 12288 Oct 15 15:27 /etc
262545 -rw-r--r-- 1 root root 2202 Oct 15 13:27 /etc/passwd
```

```
root@debian9:~# ls -ldi /dev/console /dev/sda /etc /etc/passwd
6510 crw----- 1 root root 5, 1 mars 19 11:58 /dev/console
7959 brw-rw---- 1 root disk 8, 0 mars 19 11:58 /dev/sda
131073 drwxr-xr-x 109 root root 4096 mars 19 11:59 /etc
142467 -rw-r--r-- 1 root root 1962 janv. 22 13:39 /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 :

```
root@debian9:~# cd /tmp; mkdir inode; cd inode; touch fichier1; ls -ali
total 8
262392 drwxr-xr-x 2 root root 4096 mars 19 12:32 .
262150 drwxrwxrwt 11 root root 4096 mars 19 12:32 ..
```

```
262393 -rw-r--r-- 1 root root 0 mars 19 12:32 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** :

```
262393 -rw-r--r-- 1 root root 0 mars 19 12:32 fichier1
```

Créez maintenant un lien physique :

```
# ln fichier1 fichier2 [Entrée]
```

Visualisez le résultat :

```
root@debian9:/tmp/inode# ln fichier1 fichier2
root@debian9:/tmp/inode# ls -ali
total 8
262392 drwxr-xr-x  2 root root 4096 mars 19 12:32 .
262150 drwxrwxrwt 11 root root 4096 mars 19 12:32 ..
262393 -rw-r--r--  2 root root  0 mars 19 12:32 fichier1
262393 -rw-r--r--  2 root root  0 mars 19 12:32 fichier2
```

Notez les deux lignes suivantes :

```
262393 -rw-r--r-- 2 root root 0 mars 19 12:32 fichier1
262393 -rw-r--r-- 2 root root 0 mars 19 12:32 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 :

```
root@debian9:/tmp/inode# ln -s fichier1 fichier3
root@debian9:/tmp/inode# ls -ali
total 8
262392 drwxr-xr-x  2 root root 4096 mars  19 12:33 .
262150 drwxrwxrwt 11 root root 4096 mars  19 12:32 ..
262393 -rw-r--r--   2 root root    0 mars  19 12:32 fichier1
262393 -rw-r--r--   2 root root    0 mars  19 12:32 fichier2
262396 lrwxrwxrwx   1 root root    8 mars  19 12:33 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.

<html> <div align="center"> Copyright © 2020 Hugh Norris. </div> </html>

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

Permanent link:  
<https://ittraining.team/doku.php?id=elearning:workbooks:debian:6:utilisateur:l101>

Last update: **2020/07/30 10:02**

