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

LUF101 - 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 :

```
trainee@ubuntu1604:~$ sudo su -
[sudo] password for trainee: trainee
root@ubuntu1604:~# cd /; ls -l
total 100
drwxr-xr-x   2 root root  4096 sept. 29 10:19 bin
drwxr-xr-x   3 root root  4096 sept. 29 10:20 boot
drwxrwxr-x   2 root root  4096 mai    3 07:25 cdrom
drwxr-xr-x  19 root root  4180 sept. 30 16:36 dev
drwxr-xr-x 129 root root 12288 sept. 29 10:20 etc
drwxr-xr-x   3 root root  4096 mai    3 07:27 home
lrwxrwxrwx   1 root root    32 sept. 28 10:40 initrd.img -> boot/initrd.img-4.4.0-38-generic
lrwxrwxrwx   1 root root    32 mai    3 07:31 initrd.img.old -> boot/initrd.img-4.4.0-21-generic
drwxr-xr-x  22 root root  4096 mai    3 07:47 lib
drwxr-xr-x   2 root root  4096 avril 21 00:07 lib64
drwx------   2 root root 16384 mai    3 07:17 lost+found
drwxr-xr-x   2 root root  4096 avril 21 00:07 media
```

```
drwxr-xr-x   2 root root  4096 avril 21 00:07 mnt
drwxr-xr-x   3 root root  4096 mai    3 08:14 opt
dr-xr-xr-x 123 root root     0 sept. 28 10:31 proc
drwx------   4 root root  4096 mai    3 08:33 root
drwxr-xr-x  26 root root   920 sept. 30 16:36 run
drwxr-xr-x   2 root root 12288 sept. 29 10:19 sbin
drwxr-xr-x   2 root root  4096 avril 19 16:31 snap
drwxr-xr-x   2 root root  4096 avril 21 00:07 srv
dr-xr-xr-x  13 root root     0 sept. 30 16:37 sys
drwxrwxrwt  10 root root  4096 oct.   2 05:17 tmp
drwxr-xr-x  11 root root  4096 avril 21 00:13 usr
drwxr-xr-x  14 root root  4096 avril 21 00:19 var
lrwxrwxrwx   1 root root    29 sept. 28 10:40 vmlinuz -> boot/vmlinuz-4.4.0-38-generic
lrwxrwxrwx   1 root root    29 mai    3 07:31 vmlinuz.old -> boot/vmlinuz-4.4.0-21-generic
```

```
trainee@ubuntu1804:~$ sudo su -
[sudo] password for trainee:
root@ubuntu1804:~# cd /; ls -l
total 100
drwxr-xr-x   2 root root  4096 Aug  5  2018 bin
drwxr-xr-x   4 root root  4096 Aug  5  2018 boot
drwxrwxr-x   2 root root  4096 Aug  5  2018 cdrom
drwxr-xr-x  18 root root  4080 Mar 19 08:00 dev
drwxr-xr-x 122 root root 12288 Mar 19 09:42 etc
drwxr-xr-x   5 root root  4096 Mar 19 07:56 home
lrwxrwxrwx   1 root root    33 Aug  5  2018 initrd.img -> boot/initrd.img-4.15.0-29-generic
lrwxrwxrwx   1 root root    33 Aug  5  2018 initrd.img.old -> boot/initrd.img-4.15.0-29-generic
drwxr-xr-x  21 root root  4096 Aug  5  2018 lib
drwxr-xr-x   2 root root  4096 Jul 24  2018 lib64
drwx------   2 root root 16384 Aug  5  2018 lost+found
drwxr-xr-x   2 root root  4096 Jul 24  2018 media
drwxr-xr-x   2 root root  4096 Jul 24  2018 mnt
drwxr-xr-x   2 root root  4096 Jul 24  2018 opt
dr-xr-xr-x 226 root root     0 Mar 19 07:50 proc
```

```
drwx------   4 root root  4096 Mar 19 07:54 root
drwxr-xr-x  28 root root   900 Mar 19 09:53 run
drwxr-xr-x   2 root root 12288 Aug  5  2018 sbin
drwxr-xr-x  12 root root  4096 Mar 19 08:00 snap
drwxr-xr-x   2 root root  4096 Jul 24  2018 srv
dr-xr-xr-x  13 root root     0 Mar 19 07:50 sys
drwxrwxrwt  14 root root  4096 Mar 19 09:53 tmp
drwxr-xr-x  10 root root  4096 Jul 24  2018 usr
drwxr-xr-x  14 root root  4096 Jul 24  2018 var
lrwxrwxrwx   1 root root    30 Aug  5  2018 vmlinuz -> boot/vmlinuz-4.15.0-29-generic
```

Contenu des Répertoires

L'organisation de l'arborescence d'un système de fichiers Linux commence avec la **racine** représenté par le caractère **/**. En dessous de la racine se trouve des répertoires ayant un contenu spécifique :

Directory	Contenu
/bin	Une abréviation de ***bin****ary* 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 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.
/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.

Directory	Contenu
/root	Le home de root, l'administrateur système
/run	Remplace le répertoire **/var/run**.
/sbin	Contient des binaires, donc programmes, pour l'administration du système local.
/selinux	Contient des fichiers propres à l'implémentation de SELINUX.
/snap	Utilisé avec les nouveaux paquets logiciels **Snap**.
/srv	Contient des données pour les **s**e**rv**ices 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.
/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 :

- Le fichiers normaux sont des fichiers textes, des tableaux ou des exécutables.
- La longueur du nom de fichier, y compris son extension, est limité à 255 caractères.
- Il y a une distinction entres les majuscules et le 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 :

```
root@ubuntu1604:~# 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=230832k,nr_inodes=57708,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=50028k,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,nsroot=/)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb,nsroot=/)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory,nsroot=/)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer,nsroot=/)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset,nsroot=/)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids,nsroot=/)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio,nsroot=/)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls,net_prio,nsroot=/)
```

```
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event,nsroot=/)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices,nsroot=/)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct,nsroot=/)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs
(rw,relatime,fd=22,pgrp=1,timeout=0,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)
fusectl on /sys/fs/fuse/connections type fusectl (rw,relatime)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,size=50028k,mode=700,uid=1000,gid=1000)
```

Important : Notez que sous Ubuntu 16.04, le système de fichier par défaut est **ext4**. Veuillez consulter le cours **Gestion des Disques, des Systèmes de Fichiers et du Swap** pour plus d'informations concernant les systèmes de fichiers.

```
root@ubuntu1804:/# 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=991176k,nr_inodes=247794,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=204132k,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/unified type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)
```

```
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
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/rdma type cgroup (rw,nosuid,nodev,noexec,relatime,rdma)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs
(rw,relatime,fd=30,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=12816)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,pagesize=2M)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
mqueue on /dev/mqueue type mqueue (rw,relatime)
configfs on /sys/kernel/config type configfs (rw,relatime)
fusectl on /sys/fs/fuse/connections type fusectl (rw,relatime)
/var/lib/snapd/snaps/gnome-3-26-1604_70.snap on /snap/gnome-3-26-1604/70 type squashfs (ro,nodev,relatime,x-
gdu.hide)
/var/lib/snapd/snaps/gtk-common-themes_319.snap on /snap/gtk-common-themes/319 type squashfs
(ro,nodev,relatime,x-gdu.hide)
/var/lib/snapd/snaps/gnome-system-monitor_51.snap on /snap/gnome-system-monitor/51 type squashfs
(ro,nodev,relatime,x-gdu.hide)
/var/lib/snapd/snaps/gnome-calculator_180.snap on /snap/gnome-calculator/180 type squashfs (ro,nodev,relatime,x-
gdu.hide)
/var/lib/snapd/snaps/core_4917.snap on /snap/core/4917 type squashfs (ro,nodev,relatime,x-gdu.hide)
/var/lib/snapd/snaps/gnome-logs_37.snap on /snap/gnome-logs/37 type squashfs (ro,nodev,relatime,x-gdu.hide)
/var/lib/snapd/snaps/gnome-characters_103.snap on /snap/gnome-characters/103 type squashfs (ro,nodev,relatime,x-
gdu.hide)
/dev/sda2 on /boot type ext4 (rw,relatime,data=ordered)
/dev/sda4 on /home type ext4 (rw,relatime,data=ordered)
tmpfs on /run/user/121 type tmpfs (rw,nosuid,nodev,relatime,size=204128k,mode=700,uid=121,gid=125)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,size=204128k,mode=700,uid=1000,gid=1000)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
/var/lib/snapd/snaps/core_6531.snap on /snap/core/6531 type squashfs (ro,nodev,relatime,x-gdu.hide)
```

```
/var/lib/snapd/snaps/core18_782.snap on /snap/core18/782 type squashfs (ro,nodev,relatime,x-gdu.hide)
/var/lib/snapd/snaps/gnome-logs_57.snap on /snap/gnome-logs/57 type squashfs (ro,nodev,relatime,x-gdu.hide)
/var/lib/snapd/snaps/gnome-calculator_352.snap on /snap/gnome-calculator/352 type squashfs (ro,nodev,relatime,x-
gdu.hide)
/var/lib/snapd/snaps/gnome-system-monitor_70.snap on /snap/gnome-system-monitor/70 type squashfs
(ro,nodev,relatime,x-gdu.hide)
/var/lib/snapd/snaps/gnome-characters_206.snap on /snap/gnome-characters/206 type squashfs (ro,nodev,relatime,x-
gdu.hide)
/var/lib/snapd/snaps/gtk-common-themes_1198.snap on /snap/gtk-common-themes/1198 type squashfs
(ro,nodev,relatime,x-gdu.hide)
/var/lib/snapd/snaps/gnome-3-26-1604_82.snap on /snap/gnome-3-26-1604/82 type squashfs (ro,nodev,relatime,x-
gdu.hide)
/var/lib/snapd/snaps/gnome-3-28-1804_23.snap on /snap/gnome-3-28-1804/23 type squashfs (ro,nodev,relatime,x-
gdu.hide)
tmpfs on /run/user/1001 type tmpfs (rw,nosuid,nodev,relatime,size=204128k,mode=700,uid=1001,gid=1001)
```

Important : Notez que sous Ubuntu 18.04, le système de fichier par défaut est **ext4**. 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 :

```
root@ubuntu1804:/# mount --help

Usage:
 mount [-lhV]
 mount -a [options]
 mount [options] [--source] <source> | [--target] <directory>
```

```
 mount [options] <source> <directory>
 mount <operation> <mountpoint> [<target>]

Mount a filesystem.

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
 -i, --internal-only     don't call the mount.<type> helpers
 -l, --show-labels       show also filesystem 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
 -w, --rw, --read-write  mount the filesystem read-write (default)

 -h, --help              display this help
 -V, --version           display version

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.

```
root@ubuntu1604:~# 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=c27fce7f-cc8a-4c6f-b19b-d929a4d570f2 /               ext4    errors=remount-ro 0       1
# swap was on /dev/sda5 during installation
UUID=68f67549-63f1-4833-b792-3566455bbe95 none            swap    sw              0       0
```

```
root@ubuntu1804:/# 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=4da6a147-d72f-4716-be6c-167fbec7562b /               ext4    errors=remount-ro 0       1
# /boot was on /dev/sda2 during installation
UUID=5dd0bcf0-1361-4e27-af73-ff654eca658b /boot           ext4    defaults        0       2
# /home was on /dev/sda4 during installation
UUID=a0c3f6cc-bd0a-453a-a198-282ab51fb59c /home           ext4    defaults        0       2
# swap was on /dev/sda3 during installation
UUID=4088a336-1c49-4d94-a9ed-01a96c519eb2 none            swap    sw              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 *dump* (1 = à dumper, 0 ou vide = à ignorer)	L'ordre de vérification par *fsck* 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
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 :

```
root@ubuntu1804:/# umount --help

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

Unmount filesystems.

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
 -V, --version           display version
```

```
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 :

```
root@ubuntu1604:~# dumpe2fs /dev/sda1 | grep -i superblock
dumpe2fs 1.42.13 (17-May-2015)
  Primary superblock at 0, Group descriptors at 1-1
  Backup superblock at 32768, Group descriptors at 32769-32769
  Backup superblock at 98304, Group descriptors at 98305-98305
  Backup superblock at 163840, Group descriptors at 163841-163841
  Backup superblock at 229376, Group descriptors at 229377-229377
  Backup superblock at 294912, Group descriptors at 294913-294913
```

```
  Backup superblock at 819200, Group descriptors at 819201-819201
  Backup superblock at 884736, Group descriptors at 884737-884737
  Backup superblock at 1605632, Group descriptors at 1605633-1605633
```

```
root@ubuntu1804:/# dumpe2fs /dev/sda1 | grep -i superblock
dumpe2fs 1.44.1 (24-Mar-2018)
  Primary superblock at 0, Group descriptors at 1-24
  Backup superblock at 32768, Group descriptors at 32769-32792
  Backup superblock at 98304, Group descriptors at 98305-98328
  Backup superblock at 163840, Group descriptors at 163841-163864
  Backup superblock at 229376, Group descriptors at 229377-229400
  Backup superblock at 294912, Group descriptors at 294913-294936
  Backup superblock at 819200, Group descriptors at 819201-819224
  Backup superblock at 884736, Group descriptors at 884737-884760
  Backup superblock at 1605632, Group descriptors at 1605633-1605656
  Backup superblock at 2654208, Group descriptors at 2654209-2654232
  Backup superblock at 4096000, Group descriptors at 4096001-4096024
  Backup superblock at 7962624, Group descriptors at 7962625-7962648
  Backup superblock at 11239424, Group descriptors at 11239425-11239448
  Backup superblock at 20480000, Group descriptors at 20480001-20480024
  Backup superblock at 23887872, Group descriptors at 23887873-23887896
```

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

```
# e2fsck -f -b 32768 /dev/sda1 [Enter]
```

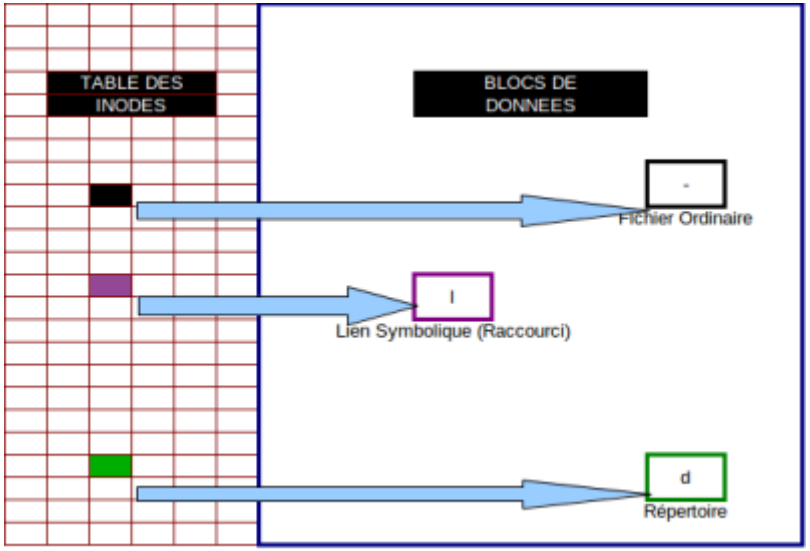
Inodes

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 **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@ubuntu1604:~# ls -ld /dev/console /dev/sda1 /etc /etc/passwd
crw-------   1 root root  5, 1 sept. 28 10:31 /dev/console
brw-rw----   1 root disk  8, 1 sept. 28 10:31 /dev/sda1
drwxr-xr-x 129 root root 12288 sept. 29 10:20 /etc
-rw-r--r--   1 root root  2296 mai    3 08:08 /etc/passwdd
```

```
root@ubuntu1804:/# ls -ld /dev/console /dev/sda1 /etc /etc/passwd
crw-------   1 root root  5, 1 Mar 19 07:51 /dev/console
brw-rw----   1 root disk  8, 1 Mar 19 07:50 /dev/sda1
drwxr-xr-x 122 root root 12288 Mar 19 09:42 /etc
-rw-r--r--   1 root root  2508 Mar 19 08:02 /etc/passwd
```

Le premier caractère de chaque ligne après le numéro de l'inode 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 **-i** :

```
root@ubuntu1604:~# ls -ldi /dev/console /dev/sda1 /etc /etc/passwd
    14 crw-------   1 root root  5, 1 sept. 28 10:31 /dev/console
   376 brw-rw----   1 root disk  8, 1 sept. 28 10:31 /dev/sda1
390913 drwxr-xr-x 129 root root 12288 sept. 29 10:20 /etc
396002 -rw-r--r--   1 root root  2296 mai    3 08:08 /etc/passwd
```

```
root@ubuntu1804:/# ls -ldi /dev/console /dev/sda1 /etc /etc/passwd
     14 crw-------   1 root root  5, 1 Mar 19 07:51 /dev/console
    350 brw-rw----   1 root disk  8, 1 Mar 19 07:50 /dev/sda1
1835009 drwxr-xr-x 122 root root 12288 Mar 19 09:42 /etc
1836859 -rw-r--r--   1 root root  2508 Mar 19 08:02 /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@ubuntu1804:/# cd /tmp; mkdir inode; cd inode; touch file1; ls -ali
total 8
1966106 drwxr-xr-x  2 root root 4096 Mar 19 10:04 .
1966081 drwxrwxrwt 15 root root 4096 Mar 19 10:04 ..
1966107 -rw-r--r--  1 root root    0 Mar 19 10:04 file1
```

file1 indique un numéro d'inode de **1966107** ainsi que le référencement d'un seul nom, indiqué par le chiffre **1** dans la troisième colonne :

```
1966107 -rw-r--r--  1 root root    0 Mar 19 10:04 file1
```

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

```
root@ubuntu1804:/tmp/inode# ln file1 file2
root@ubuntu1804:/tmp/inode# ls -ali
total 8
1966106 drwxr-xr-x  2 root root 4096 Mar 19 10:05 .
1966081 drwxrwxrwt 15 root root 4096 Mar 19 10:04 ..
1966107 -rw-r--r--  2 root root    0 Mar 19 10:04 file1
1966107 -rw-r--r--  2 root root    0 Mar 19 10:04 file2
```

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

```
1966107 -rw-r--r--  2 root root    0 Mar 19 10:04 file1
1966107 -rw-r--r--  2 root root    0 Mar 19 10:04 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 :

```
root@ubuntu1804:/tmp/inode# ln -s file1 file3
root@ubuntu1804:/tmp/inode# ls -ali
total 8
1966106 drwxr-xr-x  2 root root 4096 Mar 19 10:06 .
1966081 drwxrwxrwt 15 root root 4096 Mar 19 10:04 ..
1966107 -rw-r--r--  2 root root    0 Mar 19 10:04 file1
1966107 -rw-r--r--  2 root root    0 Mar 19 10:04 file2
```

```
1966108 lrwxrwxrwx  1 root root    5 Mar 19 10:06 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>

</html>