Version : **2022.01**

Updated : 2022/11/01 07:26

Topic 203: Filesystem and Devices

LAB #1 - 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**.

```
trainee@debian11:~$ cd /
trainee@debian11:/$ ls -l
```

```
total 60
lrwxrwxrwx   1 root root     7 Apr 25 06:26 bin -> usr/bin
drwxr-xr-x   3 root root  4096 Apr 25 06:54 boot
drwxr-xr-x  17 root root  3240 May 10 14:37 dev
drwxr-xr-x 112 root root  4096 May 10 14:37 etc
drwxr-xr-x   3 root root  4096 Apr 25 07:01 home
lrwxrwxrwx   1 root root    31 Apr 25 06:31 initrd.img -> boot/initrd.img-5.10.0-13-amd64
lrwxrwxrwx   1 root root    31 Apr 25 06:31 initrd.img.old -> boot/initrd.img-5.10.0-13-amd64
lrwxrwxrwx   1 root root     7 Apr 25 06:26 lib -> usr/lib
lrwxrwxrwx   1 root root     9 Apr 25 06:26 lib32 -> usr/lib32
lrwxrwxrwx   1 root root     9 Apr 25 06:26 lib64 -> usr/lib64
lrwxrwxrwx   1 root root    10 Apr 25 06:26 libx32 -> usr/libx32
drwx------   2 root root 16384 Apr 25 06:26 lost+found
drwxr-xr-x   3 root root  4096 Apr 25 06:26 media
drwxr-xr-x   2 root root  4096 Apr 25 06:27 mnt
drwxr-xr-x   2 root root  4096 Apr 25 06:27 opt
dr-xr-xr-x 166 root root     0 May 10 14:37 proc
drwx------   3 root root  4096 Apr 25 07:05 root
drwxr-xr-x  24 root root   660 May 10 14:37 run
lrwxrwxrwx   1 root root     8 Apr 25 06:26 sbin -> usr/sbin
drwxr-xr-x   2 root root  4096 Apr 25 06:27 srv
dr-xr-xr-x  13 root root     0 May 10 14:37 sys
drwxrwxrwt  10 root root  4096 May 10 14:37 tmp
drwxr-xr-x  14 root root  4096 Apr 25 06:27 usr
drwxr-xr-x  11 root root  4096 Apr 25 06:27 var
lrwxrwxrwx   1 root root    28 Apr 25 06:31 vmlinuz -> boot/vmlinuz-5.10.0-13-amd64
lrwxrwxrwx   1 root root    28 Apr 25 06:31 vmlinuz.old -> boot/vmlinuz-5.10.0-13-amd64
```

- **/bin** : est une abréviation de ***bin****ary* ou binaires. Il contient des programmes tels ls. Sous Debian 11 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 Debian 11 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 Debian 11 il s'agit d'un lien symbolique qui pointe vers /usr/lib64.
- **/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.
- **/sbin** : contient des binaires, donc programmes, pour l'administration du système local. Sous Debian 11 il s'agit d'un lien symbolique qui pointe vers /usr/sbin.
- **/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** : 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.

1.1 - 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 entres les majuscules et le minuscules.

Si le nom d'un fichier commence par un **.**, le fichier devient caché.

1.2 - 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@debian11:/$ su -
Password: fenestros
root@debian11:~# 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=1989872k,nr_inodes=497468,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,nodev,noexec,relatime,size=402560k,mode=755)
```

```
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro)
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)
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursiveprot)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
none on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs
(rw,relatime,fd=29,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=10999)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,pagesize=2M)
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
tmpfs on /run/user/113 type tmpfs
(rw,nosuid,nodev,relatime,size=402556k,nr_inodes=100639,mode=700,uid=113,gid=121)
tmpfs on /run/user/1000 type tmpfs
(rw,nosuid,nodev,relatime,size=402556k,nr_inodes=100639,mode=700,uid=1000,gid=1000)
```

Cette information est stockée dans le fichier **/etc/mtab** :

```
root@debian11:~# cat /etc/mtab
sysfs /sys sysfs rw,nosuid,nodev,noexec,relatime 0 0
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
udev /dev devtmpfs rw,nosuid,relatime,size=1989872k,nr_inodes=497468,mode=755 0 0
devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000 0 0
tmpfs /run tmpfs rw,nosuid,nodev,noexec,relatime,size=402560k,mode=755 0 0
/dev/sda1 / ext4 rw,relatime,errors=remount-ro 0 0
securityfs /sys/kernel/security securityfs rw,nosuid,nodev,noexec,relatime 0 0
tmpfs /dev/shm tmpfs rw,nosuid,nodev 0 0
tmpfs /run/lock tmpfs rw,nosuid,nodev,noexec,relatime,size=5120k 0 0
cgroup2 /sys/fs/cgroup cgroup2 rw,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursiveprot 0 0
pstore /sys/fs/pstore pstore rw,nosuid,nodev,noexec,relatime 0 0
```

```
none /sys/fs/bpf bpf rw,nosuid,nodev,noexec,relatime,mode=700 0 0
systemd-1 /proc/sys/fs/binfmt_misc autofs
rw,relatime,fd=29,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=10999 0 0
tracefs /sys/kernel/tracing tracefs rw,nosuid,nodev,noexec,relatime 0 0
mqueue /dev/mqueue mqueue rw,nosuid,nodev,noexec,relatime 0 0
debugfs /sys/kernel/debug debugfs rw,nosuid,nodev,noexec,relatime 0 0
hugetlbfs /dev/hugepages hugetlbfs rw,relatime,pagesize=2M 0 0
configfs /sys/kernel/config configfs rw,nosuid,nodev,noexec,relatime 0 0
fusectl /sys/fs/fuse/connections fusectl rw,nosuid,nodev,noexec,relatime 0 0
tmpfs /run/user/113 tmpfs rw,nosuid,nodev,relatime,size=402556k,nr_inodes=100639,mode=700,uid=113,gid=121 0 0
tmpfs /run/user/1000 tmpfs rw,nosuid,nodev,relatime,size=402556k,nr_inodes=100639,mode=700,uid=1000,gid=1000 0 0
```

et aussi dans le fichier **/proc/mounts** :

```
root@debian11:~# cat /proc/mounts
sysfs /sys sysfs rw,nosuid,nodev,noexec,relatime 0 0
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
udev /dev devtmpfs rw,nosuid,relatime,size=8173092k,nr_inodes=2043273,mode=755 0 0
devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000 0 0
tmpfs /run tmpfs rw,nosuid,nodev,noexec,relatime,size=1639204k,mode=755 0 0
/dev/sda1 / ext4 rw,relatime,errors=remount-ro 0 0
securityfs /sys/kernel/security securityfs rw,nosuid,nodev,noexec,relatime 0 0
tmpfs /dev/shm tmpfs rw,nosuid,nodev 0 0
tmpfs /run/lock tmpfs rw,nosuid,nodev,noexec,relatime,size=5120k 0 0
cgroup2 /sys/fs/cgroup cgroup2 rw,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursiveprot 0 0
pstore /sys/fs/pstore pstore rw,nosuid,nodev,noexec,relatime 0 0
none /sys/fs/bpf bpf rw,nosuid,nodev,noexec,relatime,mode=700 0 0
systemd-1 /proc/sys/fs/binfmt_misc autofs
rw,relatime,fd=29,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=15277 0 0
debugfs /sys/kernel/debug debugfs rw,nosuid,nodev,noexec,relatime 0 0
fusectl /sys/fs/fuse/connections fusectl rw,nosuid,nodev,noexec,relatime 0 0
mqueue /dev/mqueue mqueue rw,nosuid,nodev,noexec,relatime 0 0
tracefs /sys/kernel/tracing tracefs rw,nosuid,nodev,noexec,relatime 0 0
hugetlbfs /dev/hugepages hugetlbfs rw,relatime,pagesize=2M 0 0
```

```
configfs /sys/kernel/config configfs rw,nosuid,nodev,noexec,relatime 0 0
tmpfs /run/user/1000 tmpfs rw,nosuid,nodev,relatime,size=1639200k,nr_inodes=409800,mode=700,uid=1000,gid=1000 0 0
binfmt_misc /proc/sys/fs/binfmt_misc binfmt_misc rw,nosuid,nodev,noexec,relatime 0 0
tmpfs /run/user/113 tmpfs rw,nosuid,nodev,relatime,size=1639200k,nr_inodes=409800,mode=700,uid=113,gid=121 0 0
```

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

```
root@debian11:~# 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
     --options-mode <mode>
                         what to do with options loaded from fstab
     --options-source <source>
                         mount options source
     --options-source-force
                         force use of options from fstab/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
     --target-prefix <path>
                         specifies path use for all mountpoints
 -v, --verbose           say what is being done
 -w, --rw, --read-write  mount the filesystem read-write (default)
 -N, --namespace <ns>    perform mount in another namespace

 -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
 ID=<id>                 specifies device by udev hardware ID
 <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).
```

1.3 - La Commande umount

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

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

ou

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

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

```
root@debian11:~# 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
 -q, --quiet             suppress 'not mounted' error messages
 -N, --namespace <ns>    perform umount in another namespace

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

For more details see umount(8).
```

1.4 - 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@debian11:~# 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).
#
# systemd generates mount units based on this file, see systemd.mount(5).
```

```
# Please run 'systemctl daemon-reload' after making changes here.
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/sda1 during installation
UUID=9887a74f-a680-4bde-8f04-db5ae9ea186e /               ext4    errors=remount-ro 0       1
# swap was on /dev/sda5 during installation
UUID=1f9439f5-4b19-49b1-b292-60c2c674cee9 none            swap    sw              0       0
/dev/sr0        /media/cdrom0   udf,iso9660 user,noauto     0       0
```

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 ou une étiquette	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

Option	Systèmes de Fichier	Description	Valeur par Défaut
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

Périphériques de stockage

Les unités de stockage sous Linux sont référencées par un des fichiers se trouvant dans le répertoire **/dev** :

- hd[a-d]
  - Les disques IDE et les lecteurs ATAPI
- sd[a-z]
  - Les disques SCSI et SATA
- mmcblk[0-7]
  - Les cartes SD/MMC
- scd[0-7]
  - Les CDRoms SCSI
- xd[a-d]
  - Les premiers disques sur IBM XT
- fd[0-7]

    - Les lecteurs de disquettes
- st[0-7]
    - Les lecteurs de bandes SCSI/SATA qui **supportent** le rembobinage
- nst[0-7]
    - Les lecteurs de bandes SCSI/SATA qui ne supportent **pas** le rembobinage
- ht[0-7]
    - Les lecteurs de bandes PATA qui **supportent** le rembobinage
- nht[0-7]
    - Les lecteurs de bandes PATA qui ne supportent **pas** le rembobinage
- rmt8, rmt16, tape-d, tape-reset
    - Les lecteurs QIC-80
- ram[0-15]
    - Les disques virtuels. Ils sont supprimés à l'extinction de la machine. Un de ces disques est utilisé par le système pour monter l'image d'un disque racine défini par le fichier **initrd** au démarrage de la machine
- Périphériques **loop**
    - Il existe 16 unités loop qui sont utilisés pour accèder en mode bloc à un système de fichiers contenu dans un fichier, par exemple, une image **iso**
- md[x]
    - Un volume **RAID** logiciel
- vg[x]
    - Un groupe de volumes
- lv[x]
    - Un volume logique

Partitions

Un PC comportent en règle générale 2 **contrôleurs** de disque, chacun capable de gérer 2 disques, un **maître** et un **esclave**. Les disques attachés à ces contrôleurs comportent des noms différents pour pouvoir les distinguer :

- Contrôleur 0
    - Maître
        - **hda** - disque IDE
        - **sda** - disque SATA ou SCSI

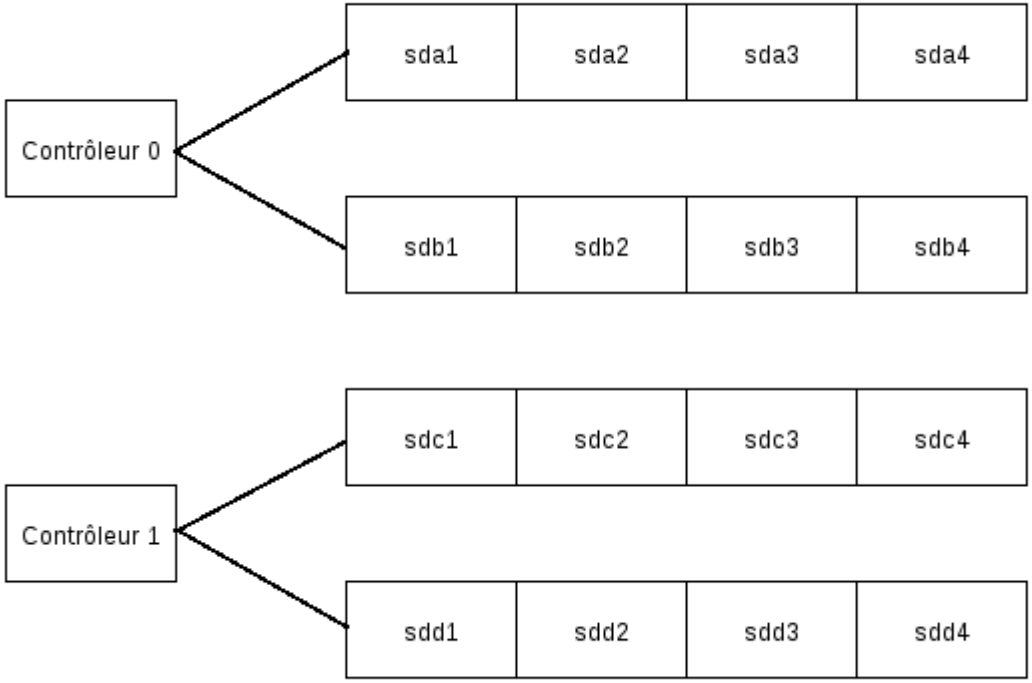
    - Esclave
        - **hdb** - disque IDE
        - **sdb** - disque SATA ou SCSI
- Contrôleur 1
    - Maître
        - **hdc** - disque IDE
        - **sdc** - disque SATA ou SCSI
    - Esclave
        - **hdd** - disque IDE
        - **sdd** - disque SATA ou SCSI

Un disque peut comporter trois types de partitions :

- **Partitions primaires**,
    - Maximum de **4**. En effet, la Table des Partitions est grande de 64 octets. Il faut 16 octets pour codés une partition.
- **Partitions Etendues**,
    - Généralement une seule partition étendue par disque. Elle contient des **Lecteurs Logiques** aussi appelés des partitions,
- **Lecteurs Logiqiques**.

Les 4 partitions primaires sont numérotées de 1 à 4. Par exemple :

- **hda1**, **hda2**, **hda3** et **hda4** pour le premier disque **IDE** sur le premier contrôleur de disque,
- **sda1**, **sda2**, **sda3** et **sda4** pour le premier disque **SCSI** ou **SATA** sur le premier contrôleur de disque.



Une partition étendue prend la place d'une partition primaire et les lecteurs logiques qui s'y trouvent commencent à partir de **hda5** ou de **sda5**.

Pour clarifier ceci, considérons un disque **SATA** contenant deux partitions primaires, une seule partition étendue et 3 lecteurs logiques. Dans ce cas, les deux premières partitions sont **sda1** et **sda2**, la partition étendue prend la place de la troisième partition primaire, la **sda3** et s'appelle ainsi tandis que la quatrième partition primaire est inexistante.

Les lecteurs logiques commençant à **sda5**, nous obtenons la liste de partitions suivante : sda1, sda2, sda5, sda6, sda7. Notez que la sda3 ne peut pas être utilisée en tant que partition car elle est cachée par les lecteurs sda5, sda6 et sda7.

Le nombre de partitions sur un disque est limité :

- **IDE**,
  - Jusqu'à **63**,
- **SCSI**,
  - Jusqu'à **15**,
- **Disques utilisant l'API libata**,
  - Jusqu'à **15**.

Important : Ces limites peuvent être dépassées en utilisant la gestion **LVM** (*Logical Volume Management*).

Partitionnement

LAB #1 - Partitionnement de votre Disque avec fdisk

Pour procéder au partitionnement de votre disque ou de vos disques, Debian 11 possède l'outil dénommé **fdisk**.

Lancez fdisk en fournissant en argument le fichier de référence de votre premier disque dur, par exemple :

```
root@debian11:~# fdisk /dev/sdc

Welcome to fdisk (util-linux 2.36.1).
```

```
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x304308a3.

Command (m for help):
```

Tapez ensuite la lettre **m** puis ↵ Entrée pour obtenir le menu :

```
Command (m for help): m

Help:

  DOS (MBR)
   a   toggle a bootable flag
   b   edit nested BSD disklabel
   c   toggle the dos compatibility flag

  Generic
   d   delete a partition
   F   list free unpartitioned space
   l   list known partition types
   n   add a new partition
   p   print the partition table
   t   change a partition type
   v   verify the partition table
   i   print information about a partition

  Misc
   m   print this menu
   u   change display/entry units
   x   extra functionality (experts only)
```

```
  Script
   I   load disk layout from sfdisk script file
   O   dump disk layout to sfdisk script file

  Save & Exit
   w   write table to disk and exit
   q   quit without saving changes

  Create a new label
   g   create a new empty GPT partition table
   G   create a new empty SGI (IRIX) partition table
   o   create a new empty DOS partition table
   s   create a new empty Sun partition table


Command (m for help):
```

Pour créer une nouvelle partition, vous devez utiliser la commande **n**.

Créez donc les partitions suivantes sur votre disque :

Partition	**Type**	**Taille de la Partition**
/dev/sdc1	Primaire	100 Mo
/dev/sdc2	Primaire	100 Mo
/dev/sdc3	Primaire	100 Mo
/dev/sdc4	Extended	Du premier secteur disponible au dernier secteur du disque
/dev/sdc5	Logique	500 Mo
/dev/sdc6	Logique	200 Mo
/dev/sdc7	Logique	300 Mo
/dev/sdc8	Logique	500 Mo
/dev/sdc9	Logique	400 Mo
/dev/sdc10	Logique	500 Mo
/dev/sdc11	Logique	500 Mo

Partition	Type	Taille de la Partition
/dev/sdc12	Logique	200 Mo

Créez d'abord les partitions primaires :

```
Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (1-4, default 1):
First sector (2048-8388607, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-8388607, default 8388607): +100M

Created a new partition 1 of type 'Linux' and of size 100 MiB.

Command (m for help): n
Partition type
   p   primary (1 primary, 0 extended, 3 free)
   e   extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (2-4, default 2):
First sector (206848-8388607, default 206848):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (206848-8388607, default 8388607): +100M

Created a new partition 2 of type 'Linux' and of size 100 MiB.

Command (m for help): n
Partition type
   p   primary (2 primary, 0 extended, 2 free)
```

```
   e   extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (3,4, default 3):
First sector (411648-8388607, default 411648):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (411648-8388607, default 8388607): +100M

Created a new partition 3 of type 'Linux' and of size 100 MiB.

Command (m for help):
```

Créez ensuite la partition étendue :

```
Command (m for help): n
Partition type
   p   primary (3 primary, 0 extended, 1 free)
   e   extended (container for logical partitions)
Select (default e):

Using default response e.
Selected partition 4
First sector (616448-8388607, default 616448):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (616448-8388607, default 8388607):

Created a new partition 4 of type 'Extended' and of size 3.7 GiB.

Command (m for help):
```

Créez maintenant les autres partitions l'une après l'autre :

```
Command (m for help): n
All primary partitions are in use.
Adding logical partition 5
```

```
First sector (618496-8388607, default 618496):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (618496-8388607, default 8388607): +500M

Created a new partition 5 of type 'Linux' and of size 500 MiB.

Command (m for help): n
All primary partitions are in use.
Adding logical partition 6
First sector (1644544-8388607, default 1644544):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (1644544-8388607, default 8388607): +200M

Created a new partition 6 of type 'Linux' and of size 200 MiB.

Command (m for help): n

All primary partitions are in use.
Adding logical partition 7
First sector (2056192-8388607, default 2056192):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2056192-8388607, default 8388607): +300M

Created a new partition 7 of type 'Linux' and of size 300 MiB.

Command (m for help): n
All primary partitions are in use.
Adding logical partition 8
First sector (2672640-8388607, default 2672640):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2672640-8388607, default 8388607): +500M

Created a new partition 8 of type 'Linux' and of size 500 MiB.

Command (m for help): n
All primary partitions are in use.
Adding logical partition 9
First sector (3698688-8388607, default 3698688):
```

```
Last sector, +/-sectors or +/-size{K,M,G,T,P} (3698688-8388607, default 8388607): +400M

Created a new partition 9 of type 'Linux' and of size 400 MiB.

Command (m for help): n
All primary partitions are in use.
Adding logical partition 10
First sector (4519936-8388607, default 4519936):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (4519936-8388607, default 8388607): +500M

Created a new partition 10 of type 'Linux' and of size 500 MiB.

Command (m for help): n
All primary partitions are in use.
Adding logical partition 11
First sector (5545984-8388607, default 5545984):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (5545984-8388607, default 8388607): +500M

Created a new partition 11 of type 'Linux' and of size 500 MiB.

Command (m for help): n
All primary partitions are in use.
Adding logical partition 12
First sector (6572032-8388607, default 6572032):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (6572032-8388607, default 8388607): +200M

Created a new partition 12 of type 'Linux' and of size 200 MiB.

Command (m for help):
```

Tapez ensuite la lettre **p** puis ↵ Entrée pour visualiser la nouvelle table des partitions. Vous obtiendrez un résultat similaire à celui-ci :

```
Command (m for help): p
Disk /dev/sdc: 4 GiB, 4294967296 bytes, 8388608 sectors
```

```
Disk model: QEMU HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x304308a3

Device     Boot   Start     End Sectors  Size Id Type
/dev/sdc1          2048  206847  204800  100M 83 Linux
/dev/sdc2        206848  411647  204800  100M 83 Linux
/dev/sdc3        411648  616447  204800  100M 83 Linux
/dev/sdc4        616448 8388607 7772160  3.7G  5 Extended
/dev/sdc5        618496 1642495 1024000  500M 83 Linux
/dev/sdc6       1644544 2054143  409600  200M 83 Linux
/dev/sdc7       2056192 2670591  614400  300M 83 Linux
/dev/sdc8       2672640 3696639 1024000  500M 83 Linux
/dev/sdc9       3698688 4517887  819200  400M 83 Linux
/dev/sdc10      4519936 5543935 1024000  500M 83 Linux
/dev/sdc11      5545984 6569983 1024000  500M 83 Linux
/dev/sdc12      6572032 6981631  409600  200M 83 Linux

Command (m for help):
```

Ecrivez la table des partitions sur disque et exécutez la commande **partprobe** :

```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

root@debian11:~# partprobe
Warning: Unable to open /dev/sr0 read-write (Read-only file system).  /dev/sr0 has been opened read-only.
```

Lancez fdisk puis tapez ensuite la lettre **p** puis ↵ Entrée pour visualiser la table des partitions actuelle :

```
root@debian11:~# fdisk /dev/sdc

Welcome to fdisk (util-linux 2.36.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): p
Disk /dev/sdc: 4 GiB, 4294967296 bytes, 8388608 sectors
Disk model: QEMU HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x304308a3

Device     Boot   Start     End Sectors  Size Id Type
/dev/sdc1          2048  206847  204800  100M 83 Linux
/dev/sdc2        206848  411647  204800  100M 83 Linux
/dev/sdc3        411648  616447  204800  100M 83 Linux
/dev/sdc4        616448 8388607 7772160  3.7G  5 Extended
/dev/sdc5        618496 1642495 1024000  500M 83 Linux
/dev/sdc6       1644544 2054143  409600  200M 83 Linux
/dev/sdc7       2056192 2670591  614400  300M 83 Linux
/dev/sdc8       2672640 3696639 1024000  500M 83 Linux
/dev/sdc9       3698688 4517887  819200  400M 83 Linux
/dev/sdc10      4519936 5543935 1024000  500M 83 Linux
/dev/sdc11      5545984 6569983 1024000  500M 83 Linux
/dev/sdc12      6572032 6981631  409600  200M 83 Linux

Command (m for help):
```

Pour supprimer une partition, utilisez la commande **d** puis ↵ Entrée. fdisk vous demandera le numéro de la partition à supprimer, par exemple :

```
Command (m for help): d
Partition number (1-12, default 12):

Partition 12 has been deleted.

Command (m for help): p
Disk /dev/sdc: 4 GiB, 4294967296 bytes, 8388608 sectors
Disk model: QEMU HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x304308a3

Device     Boot   Start     End Sectors  Size Id Type
/dev/sdc1          2048  206847  204800  100M 83 Linux
/dev/sdc2        206848  411647  204800  100M 83 Linux
/dev/sdc3        411648  616447  204800  100M 83 Linux
/dev/sdc4        616448 8388607 7772160  3.7G  5 Extended
/dev/sdc5        618496 1642495 1024000  500M 83 Linux
/dev/sdc6       1644544 2054143  409600  200M 83 Linux
/dev/sdc7       2056192 2670591  614400  300M 83 Linux
/dev/sdc8       2672640 3696639 1024000  500M 83 Linux
/dev/sdc9       3698688 4517887  819200  400M 83 Linux
/dev/sdc10      4519936 5543935 1024000  500M 83 Linux
/dev/sdc11      5545984 6569983 1024000  500M 83 Linux

Command (m for help):
```

A ce stade, la partition n'a **pas** été réellement supprimée. En effet, vous avez la possibilité de sortir de fdisk en utilisant la commande **q**.

Tapez donc q pour sortir de fdisk puis relancez fdisk. Vous obtiendrez un résultat similaire à celui-ci :

```
Command (m for help): q
```

```
root@debian11:~# fdisk /dev/sdc

Welcome to fdisk (util-linux 2.36.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): p
Disk /dev/sdc: 4 GiB, 4294967296 bytes, 8388608 sectors
Disk model: QEMU HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x304308a3

Device     Boot   Start     End Sectors  Size Id Type
/dev/sdc1          2048  206847  204800  100M 83 Linux
/dev/sdc2        206848  411647  204800  100M 83 Linux
/dev/sdc3        411648  616447  204800  100M 83 Linux
/dev/sdc4        616448 8388607 7772160  3.7G  5 Extended
/dev/sdc5        618496 1642495 1024000  500M 83 Linux
/dev/sdc6       1644544 2054143  409600  200M 83 Linux
/dev/sdc7       2056192 2670591  614400  300M 83 Linux
/dev/sdc8       2672640 3696639 1024000  500M 83 Linux
/dev/sdc9       3698688 4517887  819200  400M 83 Linux
/dev/sdc10      4519936 5543935 1024000  500M 83 Linux
/dev/sdc11      5545984 6569983 1024000  500M 83 Linux
/dev/sdc12      6572032 6981631  409600  200M 83 Linux

Command (m for help):
```

LAB #2 - Modifier les Drapeaux des Partitions avec fdisk

Afin de mettre en place un RAID logiciel ou un volume logique, il est nécessaire de modifier les types de systèmes de fichiers sur les partitions créées.

Modifiez donc les nouvelles partitions à l'aide de la commande **t** de **fdisk** selon le tableau ci-dessous :

Taille de la Partition	Type de Système de Fichiers
500 Mo	RAID (fd)
200 Mo	Linux LVM (8e)
300 Mo	Linux LVM (8e)
500 Mo	RAID (fd)
400 Mo	Linux LVM (8e)
500 Mo	RAID (fd)
500 Mo	RAID (fd)
200 Mo	Inchangé

Vous obtiendrez un résultat similaire à celui-ci :

```
Command (m for help): t
Partition number (1-12, default 12): 5
Hex code or alias (type L to list all): fd

Changed type of partition 'Linux' to 'Linux raid autodetect'.

Command (m for help): t
Partition number (1-12, default 12): 6
Hex code or alias (type L to list all): 8e

Changed type of partition 'Linux' to 'Linux LVM'.

Command (m for help): t
Partition number (1-12, default 12): 7
Hex code or alias (type L to list all): 8e
```

```
Changed type of partition 'Linux' to 'Linux LVM'.

Command (m for help): t
Partition number (1-12, default 12): 8
Hex code or alias (type L to list all): fd

Changed type of partition 'Linux' to 'Linux raid autodetect'.

Command (m for help): t
Partition number (1-12, default 12): 9
Hex code or alias (type L to list all): 8e

Changed type of partition 'Linux' to 'Linux LVM'.

Command (m for help): t
Partition number (1-12, default 12): 10
Hex code or alias (type L to list all): fd

Changed type of partition 'Linux' to 'Linux raid autodetect'.

Command (m for help): t
Partition number (1-12, default 12): 11
Hex code or alias (type L to list all): fd

Changed type of partition 'Linux' to 'Linux raid autodetect'.

Command (m for help):
```

A l'issu des modifications, vous obtiendrez un résultat similaire à celui-ci :

```
Command (m for help): p
Disk /dev/sdc: 4 GiB, 4294967296 bytes, 8388608 sectors
Disk model: QEMU HARDDISK
Units: sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x304308a3

Device     Boot   Start     End Sectors  Size Id Type
/dev/sdc1          2048  206847  204800  100M 83 Linux
/dev/sdc2        206848  411647  204800  100M 83 Linux
/dev/sdc3        411648  616447  204800  100M 83 Linux
/dev/sdc4        616448 8388607 7772160  3.7G  5 Extended
/dev/sdc5        618496 1642495 1024000  500M fd Linux raid autodetect
/dev/sdc6       1644544 2054143  409600  200M 8e Linux LVM
/dev/sdc7       2056192 2670591  614400  300M 8e Linux LVM
/dev/sdc8       2672640 3696639 1024000  500M fd Linux raid autodetect
/dev/sdc9       3698688 4517887  819200  400M 8e Linux LVM
/dev/sdc10      4519936 5543935 1024000  500M fd Linux raid autodetect
/dev/sdc11      5545984 6569983 1024000  500M fd Linux raid autodetect
/dev/sdc12      6572032 6981631  409600  200M 83 Linux

Command (m for help):
```

Pour écrire la nouvelle table des partitions sur disque, vous devez utilisez la commande **w** puis la commande **partprobe** :

```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

root@debian11:~# partprobe
Warning: Unable to open /dev/sr0 read-write (Read-only file system).  /dev/sr0 has been opened read-only.
```

Options de la Commande fdisk

Les options de cette commande sont :

```
root@debian11:~# fdisk --help

Usage:
 fdisk [options] <disk>         change partition table
 fdisk [options] -l [<disk>...] list partition table(s)

Display or manipulate a disk partition table.

Options:
 -b, --sector-size <size>      physical and logical sector size
 -B, --protect-boot            don't erase bootbits when creating a new label
 -c, --compatibility[=<mode>]  mode is 'dos' or 'nondos' (default)
 -L, --color[=<when>]          colorize output (auto, always or never)
                                 colors are enabled by default
 -l, --list                    display partitions and exit
 -x, --list-details            like --list but with more details
 -n, --noauto-pt               don't create default partition table on empty devices
 -o, --output <list>           output columns
 -t, --type <type>             recognize specified partition table type only
 -u, --units[=<unit>]          display units: 'cylinders' or 'sectors' (default)
 -s, --getsz                   display device size in 512-byte sectors [DEPRECATED]
     --bytes                   print SIZE in bytes rather than in human readable format
     --lock[=<mode>]           use exclusive device lock (yes, no or nonblock)
 -w, --wipe <mode>             wipe signatures (auto, always or never)
 -W, --wipe-partitions <mode>  wipe signatures from new partitions (auto, always or never)

 -C, --cylinders <number>      specify the number of cylinders
 -H, --heads <number>          specify the number of heads
 -S, --sectors <number>        specify the number of sectors per track
```

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

Available output columns:
 gpt: Device Start End Sectors Size Type Type-UUID Attrs Name UUID
 dos: Device Start End Sectors Cylinders Size Type Id Attrs Boot End-C/H/S Start-C/H/S
 bsd: Slice Start End Sectors Cylinders Size Type Bsize Cpg Fsize
 sgi: Device Start End Sectors Cylinders Size Type Id Attrs
 sun: Device Start End Sectors Cylinders Size Type Id Flags

For more details see fdisk(8).
```

Systèmes de Fichiers Journalisés

Présentation

Un journal est la partie d'un système de fichiers journalisé qui trace les opérations d'écriture tant qu'elles ne sont pas terminées et cela en vue de garantir l'intégrité des données en cas d'arrêt brutal.

L'intérêt est de pouvoir plus facilement et plus rapidement récupérer les données en cas d'arrêt brutal du système d'exploitation (coupure d'alimentation, plantage du système, etc.), alors que les partitions n'ont pas été correctement synchronisées et démontées.

Sans un tel fichier journal, un outil de récupération de données après un arrêt brutal doit parcourir l'intégralité du système de fichier pour vérifier sa cohérence. Lorsque la taille du système de fichiers est importante, cela peut durer très longtemps pour un résultat moins efficace car entraînant des pertes de données.

Linux peut utiliser un des systèmes de fichiers journalisés suivants :

Système de fichier	Taille maximum - fichier	Taille maximum - système de fichier
Ext3	2 To	32 To
Ext4	16 To	1 EiB
XFS	8 EiB	16 EiB

Système de fichier	Taille maximum - fichier	Taille maximum - système de fichier
ReiserFS v3	8 To	16 To
JFS	4 Po	32 Po
Btrfs	16 EiB	16 EiB

A faire : Pour comparer ces six systèmes de fichier, veuillez consulter **cette page**

Ext3

Ext3 est une évolution de Ext2 et a pour principale différence d'utiliser un fichier journal. Il peut :

- être utilisé à partir d'une partition Ext2, sans avoir à sauvegarder et à restaurer des données,
- utiliser tous les utilitaires de maintenance pour les systèmes de fichiers ext2, comme fsck,
- utiliser le logiciel dump, ce qui n'est pas le cas avec ReiserFS.

Pour plus d'information concernant Ext3, consultez **cette page**

LAB# 3 - Gestion d'Ext3

Notez maintenant le numéro de la dernière partition que vous avez précédemment créée :

```
root@debian11:~# fdisk -l
Disk /dev/sda: 64 GiB, 68719476736 bytes, 134217728 sectors
Disk model: QEMU HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disk /dev/sdc: 4 GiB, 4294967296 bytes, 8388608 sectors
Disk model: QEMU HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x304308a3

Device     Boot   Start     End Sectors  Size Id Type
/dev/sdc1          2048  206847  204800  100M 83 Linux
/dev/sdc2        206848  411647  204800  100M 83 Linux
/dev/sdc3        411648  616447  204800  100M 83 Linux
/dev/sdc4        616448 8388607 7772160  3.7G  5 Extended
/dev/sdc5        618496 1642495 1024000  500M fd Linux raid autodetect
/dev/sdc6       1644544 2054143  409600  200M 8e Linux LVM
/dev/sdc7       2056192 2670591  614400  300M 8e Linux LVM
/dev/sdc8       2672640 3696639 1024000  500M fd Linux raid autodetect
/dev/sdc9       3698688 4517887  819200  400M 8e Linux LVM
/dev/sdc10      4519936 5543935 1024000  500M fd Linux raid autodetect
/dev/sdc11      5545984 6569983 1024000  500M fd Linux raid autodetect
/dev/sdc12      6572032 6981631  409600  200M 83 Linux


Disk /dev/sdb: 32 GiB, 34359738368 bytes, 67108864 sectors
Disk model: QEMU HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xf2e3a71a

Device     Boot    Start      End  Sectors  Size Id Type
/dev/sdb1  *        2048 65107967 65105920   31G 83 Linux
/dev/sdb2       65110014 67106815  1996802  975M  5 Extended
```

```
/dev/sdb5       65110016 67106815  1996800  975M 82 Linux swap / Solaris


Disk /dev/mapper/vg0-lv1: 104 MiB, 109051904 bytes, 212992 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/mapper/vg0-lv2: 112 MiB, 117440512 bytes, 229376 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 65536 bytes / 131072 bytes
```

Dans ce cas, il s'agit de **/dev/sdc12**.

Créez un filesystem Ext3 sur /dev/sdc12 en utilisant la commande **mke2fs -j** :

```
root@debian11:~# mke2fs -j /dev/sdc12
mke2fs 1.46.2 (28-Feb-2021)
Discarding device blocks: done
Creating filesystem with 204800 1k blocks and 51200 inodes
Filesystem UUID: bdea8b16-8d0b-480c-ab37-68d7d2bdb0b8
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

Les options de la commande **mke2fs** sont :

```
root@debian11:~# mke2fs --help
```

```
mke2fs: invalid option -- '-'
Usage: mke2fs [-c|-l filename] [-b block-size] [-C cluster-size]
        [-i bytes-per-inode] [-I inode-size] [-J journal-options]
        [-G flex-group-size] [-N number-of-inodes] [-d root-directory]
        [-m reserved-blocks-percentage] [-o creator-os]
        [-g blocks-per-group] [-L volume-label] [-M last-mounted-directory]
        [-O feature[,...]] [-r fs-revision] [-E extended-option[,...]]
        [-t fs-type] [-T usage-type ] [-U UUID] [-e errors_behavior][-z undo_file]
        [-jnqvDFSV] device [blocks-count]
```

Important : Lors de la mise en place d'un filesystem ext2/ext3/ext4, le système réserve 5% de l'espace disque pour root. Sur des disques de grande taille il est parfois préférable de récupérer une partie de cet espace en utilisant la commande **tune2fs -m n /dev/sdXY** ou n est le nouveau pourcentage à réserver.

LAB #4 - Convertir un Système de Fichiers Ext3 en Ext2

Pour vérifier si un système de fichiers Ext2 est journalisé, utilisez la commande **dumpe2fs** :

```
root@debian11:~# dumpe2fs -h /dev/sdc12
dumpe2fs 1.46.2 (28-Feb-2021)
Filesystem volume name:   <none>
Last mounted on:          <not available>
Filesystem UUID:          bdea8b16-8d0b-480c-ab37-68d7d2bdb0b8
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index filetype sparse_super large_file
Filesystem flags:         signed_directory_hash
Default mount options:    user_xattr acl
Filesystem state:         clean
Errors behavior:          Continue
```

```
Filesystem OS type:       Linux
Inode count:              51200
Block count:              204800
Reserved block count:     10240
Overhead clusters:        12095
Free blocks:              192674
Free inodes:              51189
First block:              1
Block size:               1024
Fragment size:            1024
Reserved GDT blocks:      256
Blocks per group:         8192
Fragments per group:      8192
Inodes per group:         2048
Inode blocks per group:   256
Filesystem created:       Tue Apr 26 16:41:16 2022
Last mount time:          n/a
Last write time:          Tue Apr 26 16:41:17 2022
Mount count:              0
Maximum mount count:      -1
Last checked:             Tue Apr 26 16:41:16 2022
Check interval:           0 (<none>)
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
First inode:              11
Inode size:               128
Journal inode:            8
Default directory hash:   half_md4
Directory Hash Seed:      705a3710-674b-4734-bf64-35a6ca7aabc5
Journal backup:           inode blocks
Journal features:         (none)
Total journal size:       4096k
Total journal blocks:     4096
Max transaction length:   4096
```

```
Fast commit length:       0
Journal sequence:         0x00000001
Journal start:            0
```

Important : Le drapeau **Filesystem features: has_journal ...** démontre que Ext3 est utilisé sur cette partition.

Pour supprimer Ext3 sur cette partition, il convient d'utiliser la commande **tune2fs**

```
root@debian11:~# tune2fs -O ^has_journal /dev/sdc12
tune2fs 1.46.2 (28-Feb-2021)
```

Constatez le résultat de cette commande :

```
root@debian11:~# dumpe2fs -h /dev/sdc12
dumpe2fs 1.46.2 (28-Feb-2021)
Filesystem volume name:   <none>
Last mounted on:          <not available>
Filesystem UUID:          bdea8b16-8d0b-480c-ab37-68d7d2bdb0b8
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      ext_attr resize_inode dir_index filetype sparse_super large_file
Filesystem flags:         signed_directory_hash
Default mount options:    user_xattr acl
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              51200
Block count:              204800
Reserved block count:     10240
Overhead clusters:        12095
Free blocks:              196787
```

```
Free inodes:              51189
First block:              1
Block size:               1024
Fragment size:            1024
Reserved GDT blocks:      256
Blocks per group:         8192
Fragments per group:      8192
Inodes per group:         2048
Inode blocks per group:   256
Filesystem created:       Tue Apr 26 16:41:16 2022
Last mount time:          n/a
Last write time:          Tue Apr 26 16:44:45 2022
Mount count:              0
Maximum mount count:      -1
Last checked:             Tue Apr 26 16:41:16 2022
Check interval:           0 (<none>)
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
First inode:              11
Inode size:               128
Default directory hash:   half_md4
Directory Hash Seed:      705a3710-674b-4734-bf64-35a6ca7aabc5
Journal backup:           inode blocks
```

Important : Notez que le drapeau **Filesystem features: has_journal ...** a été supprimé.

Supprimez maintenant l'inode du journal :

```
root@debian11:~# fsck /dev/sdc12
fsck from util-linux 2.36.1
e2fsck 1.46.2 (28-Feb-2021)
```

```
/dev/sdc12: clean, 11/51200 files, 8013/204800 blocks
```

Créez un point de montage pour /dev/sdc12 :

```
root@debian11:~# mkdir /mnt/sdc12
```

Essayez de monter /dev/sdc12 en tant que système de fichiers Ext3. Vous obtiendrez un résultat similaire à celui-ci :

```
root@debian11:~# mount -t ext3 /dev/sdc12 /mnt/sdc12
mount: /mnt/sdc12: wrong fs type, bad option, bad superblock on /dev/sdc12, missing codepage or helper program,
or other error.
```

Important : Notez l'erreur due au mauvais système de fichiers qui suit l'option **-t**.

Montez maintenant le système de fichiers en tant que Ext2 :

```
root@debian11:~# mkdir /mnt/sdc12
root@debian11:~# mount -t ext3 /dev/sdc12 /mnt/sdc12
mount: /mnt/sdc12: wrong fs type, bad option, bad superblock on /dev/sdc12, missing codepage or helper program,
or other error.
root@debian11:~# mount -t ext2 /dev/sdc12 /mnt/sdc12

root@debian11:~# cat /etc/mtab | grep sdc12
/dev/sdc12 /mnt/sdc12 ext2 rw,relatime 0 0
```

LAB #5 - Convertir un Système de Fichiers Ext2 en Ext3

Pour replacer le journal sur /dev/sdc12, il convient d'utiliser la commande **tune2fs** :

```
root@debian11:~# umount /mnt/sdc12

root@debian11:~# tune2fs -j /dev/sdc12
tune2fs 1.46.2 (28-Feb-2021)
Creating journal inode: done

root@debian11:~# dumpe2fs -h /dev/sdc12 | head
dumpe2fs 1.46.2 (28-Feb-2021)
Filesystem volume name:   <none>
Last mounted on:          <not available>
Filesystem UUID:          bdea8b16-8d0b-480c-ab37-68d7d2bdb0b8
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index filetype sparse_super large_file
Filesystem flags:         signed_directory_hash
Default mount options:    user_xattr acl
Filesystem state:         clean
Errors behavior:          Continue
```

Important : Notez que vous avez du démonter la partition avant d'exécuter la commande **tune2fs**.

LAB #6 - Placer le Journal sur un autre Partition

Le journal d'un système de fichiers peut être placé sur un autre périphérique bloc.

Créez un système de fichiers sur /dev/sdc11 :

```
root@debian11:~# mke2fs -O journal_dev /dev/sdc11
```

Important : Notez l'utilisation de l'option **-O**.

Créez maintenant un système de fichiers Ext3 sur /dev/sdc12 en plaçant le journal sur /dev/sdc11 :

```
root@debian11:~# mke2fs -j -J device=/dev/sdc11 /dev/sdc12
mke2fs 1.46.2 (28-Feb-2021)
Using journal device's blocksize: 1024
/dev/sdc12 contains a ext3 file system
        last mounted on Tue Apr 26 16:47:45 2022
Proceed anyway? (y,N) y
Discarding device blocks: done
Creating filesystem with 204800 1k blocks and 51200 inodes
Filesystem UUID: 84363ced-42b4-49b1-8627-49c39bd71ac3
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Adding journal to device /dev/sdc11: done
Writing superblocks and filesystem accounting information: done
```

Important : Notez que le journal a été placé sur /dev/sdc11 graĉe à l'utilisation de l'option **-J**.

LAB #7 - Modifier la Fréquence de Vérification du Système de Fichiers Ext3

Pour modifiez la fréquence de vérification du système de fichiers sur /dev/sdc12, il convient d'utiliser soit l'option **-c**, soit l'option **-i** :

```
root@debian11:~# tune2fs -i 100d /dev/sdc12
tune2fs 1.46.2 (28-Feb-2021)
Setting interval between checks to 8640000 seconds
```

Dernièrement, pour obtenir seul l'UUID du système de fichiers, utilisez les commandes **dumpe2fs** et **grep** :

```
root@debian11:~# dumpe2fs /dev/sdc12| grep UUID
dumpe2fs 1.46.2 (28-Feb-2021)
Filesystem UUID:          84363ced-42b4-49b1-8627-49c39bd71ac3
Journal UUID:             a9765f56-d8d4-499b-8fc4-db1d665cdfca
```

Ext4

Le système de fichiers **Ext4** fut introduit dans le noyau **2.6.19** en mode expérimental et est devenu stable dans le noyau **2.6.28**.

Ext4 n'est pas une évolution de Ext3. Cependant il a une compatibilité ascendante avec Ext3.

Les fonctionnalités majeures d'Ext4 sont :

- la gestion des volumes d'une taille allant jusqu'à **1 024 pébioctets**,
- l'allocation par **extents** qui permettent la pré-allocation d'une zone contiguë pour un fichier afin de minimiser la fragmentation.

L'option **extents** est activée par défaut depuis le noyau **2.6.23**.

La compatibilité ascendante avec ext3 comprend :

- la possibilité de monter une partition Ext3 en tant que partition Ext4,
- la possibilité de monter une partition Ext4 en tant que partition Ext3 mais **uniquement** dans le cas où la partition Ext4 n'ait jamais utilisé l'allocation par **extents** pour enregistrer des fichiers, mais l'allocation binaire comprise par ext3.

Pour plus d'informations concernant Ext4, consultez **cette page**.

LAB #8 - Créer un Système de Fichiers Ext4

Créez un système de fichiers Ext4 sur **/dev/sdc11** :

```
root@debian11:~# mkfs.ext4 /dev/sdc11
mke2fs 1.46.2 (28-Feb-2021)
/dev/sdc11 contains a jbd file system
Proceed anyway? (y,N) y
Discarding device blocks: done
Creating filesystem with 512000 1k blocks and 128016 inodes
Filesystem UUID: a590c318-d9ad-4883-abce-4e154c80e521
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729, 204801, 221185, 401409

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

Les options de cette commande sont :

```
root@debian11:~# mkfs.ext4 --help
mkfs.ext4: invalid option -- '-'
Usage: mkfs.ext4 [-c|-l filename] [-b block-size] [-C cluster-size]
        [-i bytes-per-inode] [-I inode-size] [-J journal-options]
        [-G flex-group-size] [-N number-of-inodes] [-d root-directory]
        [-m reserved-blocks-percentage] [-o creator-os]
        [-g blocks-per-group] [-L volume-label] [-M last-mounted-directory]
        [-O feature[,...]] [-r fs-revision] [-E extended-option[,...]]
        [-t fs-type] [-T usage-type ] [-U UUID] [-e errors_behavior][-z undo_file]
        [-jnqvDFSV] device [blocks-count]
```

Consultez maintenant les caractéristiques du système de fichier :

```
root@debian11:~# dumpe2fs /dev/sdc11 | more
dumpe2fs 1.46.2 (28-Feb-2021)
Filesystem volume name:   <none>
Last mounted on:          <not available>
Filesystem UUID:          a590c318-d9ad-4883-abce-4e154c80e521
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index filetype extent
64bit flex_bg sparse_super large_file huge_file dir_nlink extra_isize metadata_csum
Filesystem flags:         signed_directory_hash
Default mount options:    user_xattr acl
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              128016
Block count:              512000
Reserved block count:     25600
Overhead clusters:        26670
Free blocks:              485316
Free inodes:              128005
First block:              1
Block size:               1024
Fragment size:            1024
Group descriptor size:    64
Reserved GDT blocks:      256
--More--
[q]
```

LAB #9 - Ajouter une Etiquette au Système de Fichiers Ext4

Utilisez la commande **e2label** pour associer une étiquette au système de fichiers :

```
root@debian11:~# e2label /dev/sdc11 my_ext4
root@debian11:~# dumpe2fs /dev/sdc11 | more
dumpe2fs 1.46.2 (28-Feb-2021)
Filesystem volume name:   my_ext4
Last mounted on:          <not available>
Filesystem UUID:          a590c318-d9ad-4883-abce-4e154c80e521
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index filetype extent
64bit flex_bg sparse_super large_file huge_file dir_nlink extra_isize metadata_csum
Filesystem flags:         signed_directory_hash
Default mount options:    user_xattr acl
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              128016
Block count:              512000
Reserved block count:     25600
Overhead clusters:        26670
Free blocks:              485316
Free inodes:              128005
First block:              1
Block size:               1024
Fragment size:            1024
Group descriptor size:    64
Reserved GDT blocks:      256
--More--
[q]
```

Important - Notez que l'étiquette doit être de 16 caractères maximum.

Créez un point de montage dans **/mnt** et essayez de monter **/dev/sdc11** en tant qu'Ext3 :

```
root@debian11:~# mkdir /mnt/sdc11

root@debian11:~# mount -t ext3 /dev/sdc11 /mnt/sdc11
mount: /mnt/sdc11: wrong fs type, bad option, bad superblock on /dev/sdc11, missing codepage or helper program,
or other error.
```

Important - Notez l'erreur qui est signalée.

Montez de nouveau la partition **sans** stipuler le type de système de fichiers :

```
root@debian11:~# mount /dev/sdc11 /mnt/sdc11

root@debian11:~# mount | grep sdc11
/dev/sdc11 on /mnt/sdc11 type ext4 (rw,relatime)
```

Important - Constatez que la partition a été monté en tant qu'Ext4.

LAB #10 - Convertir un Système de Fichiers Ext3 en Ext4

Créez un système de fichiers ext3 sur /dev/sdc12 :

```
root@debian11:~# mkfs.ext3 /dev/sdc12
mke2fs 1.46.2 (28-Feb-2021)
/dev/sdc12 contains a ext3 file system
        created on Wed Apr 27 11:57:30 2022
```

```
Proceed anyway? (y,N) y
Discarding device blocks: done
Creating filesystem with 204800 1k blocks and 51200 inodes
Filesystem UUID: 62cb630b-25d5-4ca0-bbab-e855fba96a80
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

Montez maintenant **/dev/sdc12** sur /mnt/sdc12 :

```
root@debian11:~# mount /dev/sdc12 /mnt/sdc12

root@debian11:~# ls -l /mnt/sdc12
total 12
drwx------ 2 root root 12288 Apr 27 12:11 lost+found
```

Créez le fichier **/mnt/sdc12/check_file** :

```
root@debian11:~# touch /mnt/sdc12/check_file
```

Injectez la chaine **check file** dans le fichier /mnt/sdc12/check_file puis démontez /dev/sdc12 :

```
root@debian11:~# echo "check file" > /mnt/sdc12/check_file

root@debian11:~# umount /mnt/sdc12
```

Exécutez e2fsck sur /dev/sdc12 :

```
root@debian11:~# e2fsck /dev/sdc12
e2fsck 1.46.2 (28-Feb-2021)
```

```
/dev/sdc12: clean, 12/51200 files, 12127/204800 blocks
```

Convertissez /dev/sdc12 en Ext4 :

```
root@debian11:~# tune2fs -O extents,uninit_bg,dir_index /dev/sdc12
tune2fs 1.46.2 (28-Feb-2021)
```

Optimisez le système de fichiers :

```
root@debian11:~# e2fsck -fDC0 /dev/sdc12
e2fsck 1.46.2 (28-Feb-2021)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 3A: Optimizing directories
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/sdc12: ***** FILE SYSTEM WAS MODIFIED *****
/dev/sdc12: 12/51200 files (0.0% non-contiguous), 12127/204800 blocks
```

Essayez de monter **/dev/sdc12** en tant qu'Ext3 :

```
root@debian11:~# mount -t ext3 /dev/sdc12 /mnt/sdc12
mount: /mnt/sdc12: wrong fs type, bad option, bad superblock on /dev/sdc12, missing codepage or helper program,
or other error.
```

Montez /dev/sdc12 sans spécifier le type de fichiers Ext3 et vérifiez le contenu du fichier **check_file** :

```
root@debian11:~# mount /dev/sdc12 /mnt/sdc12

root@debian11:~# ls -l /mnt/sdc12
total 13
-rw-r--r-- 1 root root    11 Apr 27 12:15 check_file
drwx------ 2 root root 12288 Apr 27 12:11 lost+found
```

```
root@debian11:~# cat /mnt/sdc12/check_file
check file
```

Dernièrement, pour obtenir seul l'UUID du système de fichiers, utilisez les commandes **dumpe2fs** et **grep** :

```
root@debian11:~# dumpe2fs /dev/sdc11 | grep UUID
dumpe2fs 1.46.2 (28-Feb-2021)
Filesystem UUID:          a590c318-d9ad-4883-abce-4e154c80e521

root@debian11:~# dumpe2fs /dev/sdc12 | grep UUID
dumpe2fs 1.46.2 (28-Feb-2021)
Filesystem UUID:          62cb630b-25d5-4ca0-bbab-e855fba96a80
```

XFS

XFS est un système de fichiers 64-bit journalisé de haute performance crée par SGI pour son système d'exploitation IRIX. XFS est inclus par défaut avec les versions du noyau Linux 2.5.xx et 2.6.xx.

Pour plus d'informations concernant XFS, consultez **cette page**.

LAB #11 - Créer un Système de Fichiers XFS

Installez le paquet **xfsprogs** :

```
root@debian11:~# apt-get -y install xfsprogs
```

Créez un système de fichiers XFS sur la partition **/dev/sdc10** :

```
root@debian11:~# mkfs.xfs /dev/sdc10
meta-data=/dev/sdc10             isize=512    agcount=4, agsize=32000 blks
         =                       sectsz=512   attr=2, projid32bit=1
```

```
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=0
data     =                       bsize=4096   blocks=128000, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=1368, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
Discarding blocks...Done.
```

Consultez maintenant les caractéristiques du système de fichier :

```
root@debian11:~# xfs_info /dev/sdc10
meta-data=/dev/sdc10             isize=512    agcount=4, agsize=32000 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=0
data     =                       bsize=4096   blocks=128000, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=1368, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

LAB #12 - Ajouter une Etiquette au Système de Fichiers XFS

Utilisez la commande **xfs_admin** pour associer une étiquette au système de fichiers :

```
root@debian11:~# xfs_admin -L my_xfs /dev/sdc10
writing all SBs
new label = "my_xfs"
```

Pour voir l'étiquette, utilisez la commande suivante :

```
root@debian11:~# xfs_admin -l /dev/sdc10
label = "my_xfs"
```

Important - Notez que l'étiquette doit être de 12 caractères maximum.

Dernièrement, pour obtenir seul l'UUID du système de fichiers, utilisez la commande **xfs-admin** et l'option **-u** :

```
root@debian11:~# xfs_admin -u /dev/sdc10
UUID = b4d2c2c3-94bc-4408-8284-5b09ab60204e
```

Important - La commande **xfs_metadump** est utilisée pour sauvegarder les méta-données du système de fichiers, tandis que la commande **xfs_mdrestore** est utilisée pour restaurer les les méta-données du système de fichiers.

ReiserFS

ReiserFS permet :

- de meilleurs temps d'accès à des sous-répertoires que Ext3, même ceux contenant des dizaines de milliers de fichiers,
- une plus grande efficacité pour ce qui concerne le stockage des fichiers moins de quelques ko. Le gain d'espace peut aller jusqu'à 10% par rapport à Ext2/Ext3.

Pour plus d'informations concernant ReiserFS, consultez **cette page**.

Avant de continuer, installer le paquet **reiserfsprogs** :

```
root@debian11:~# apt-get -y install reiserfsprogs
```

LAB #13 - Créer un Système de Fichiers ReiserFS

Créez un système de fichiers ReiserFS sur la partition **/dev/sdc8** :

```
root@debian11:~# mkfs.reiserfs /dev/sdc8
mkfs.reiserfs 3.6.27

Guessing about desired format.. Kernel 5.10.0-13-amd64 is running.
Format 3.6 with standard journal
Count of blocks on the device: 128000
Number of blocks consumed by mkreiserfs formatting process: 8215
Blocksize: 4096
Hash function used to sort names: "r5"
Journal Size 8193 blocks (first block 18)
Journal Max transaction length 1024
inode generation number: 0
UUID: 95c81e34-2215-4b96-be47-e5368444f19a
ATTENTION: YOU SHOULD REBOOT AFTER FDISK!
        ALL DATA WILL BE LOST ON '/dev/sdc8'!
Continue (y/n):y
Initializing journal - 0%....20%....40%....60%....80%....100%
Syncing..ok
ReiserFS is successfully created on /dev/sdc8.
```

Consultez maintenant les caractéristiques du système de fichier :

```
root@debian11:~# debugreiserfs /dev/sdc8
debugreiserfs 3.6.27
```

```
Filesystem state: consistent

Reiserfs super block in block 16 on 0x828 of format 3.6 with standard journal
Count of blocks on the device: 128000
Number of bitmaps: 4
Blocksize: 4096
Free blocks (count of blocks - used [journal, bitmaps, data, reserved] blocks): 119785
Root block: 8211
Filesystem is clean
Tree height: 2
Hash function used to sort names: "r5"
Objectid map size 2, max 972
Journal parameters:
        Device [0x0]
        Magic [0x345d673e]
        Size 8193 blocks (including 1 for journal header) (first block 18)
        Max transaction length 1024 blocks
        Max batch size 900 blocks
        Max commit age 30
Blocks reserved by journal: 0
Fs state field: 0x0:
sb_version: 2
inode generation number: 0
UUID: 95c81e34-2215-4b96-be47-e5368444f19a
LABEL:
Set flags in SB:
        ATTRIBUTES CLEAN
Mount count: 1
Maximum mount count: 30
Last fsck run: Wed Apr 27 13:37:02 2022
Check interval in days: 180
```

LAB #14 - Ajouter une Etiquette au Système de Fichiers ReiserFS

Utilisez la commande **reiserfstune** pour associer une étiquette au système de fichiers :

```
root@debian11:~# reiserfstune -l my_reiserfs /dev/sdc8
reiserfstune: Journal device has not been specified. Assuming journal is on the main device (/dev/sdc8).

Current parameters:

Filesystem state: consistent

Reiserfs super block in block 16 on 0x828 of format 3.6 with standard journal
Count of blocks on the device: 128000
Number of bitmaps: 4
Blocksize: 4096
Free blocks (count of blocks - used [journal, bitmaps, data, reserved] blocks): 119785
Root block: 8211
Filesystem is clean
Tree height: 2
Hash function used to sort names: "r5"
Objectid map size 2, max 972
Journal parameters:
        Device [0x0]
        Magic [0x345d673e]
        Size 8193 blocks (including 1 for journal header) (first block 18)
        Max transaction length 1024 blocks
        Max batch size 900 blocks
        Max commit age 30
Blocks reserved by journal: 0
Fs state field: 0x0:
sb_version: 2
inode generation number: 0
UUID: 95c81e34-2215-4b96-be47-e5368444f19a
```

```
LABEL: my_reiserfs
Set flags in SB:
        ATTRIBUTES CLEAN
Mount count: 1
Maximum mount count: 30
Last fsck run: Wed Apr 27 13:37:02 2022
Check interval in days: 180
```

Important - Notez que l'étiquette doit être de 16 caractères maximum.

Dernièrement, pour obtenir l'UUID du système de fichiers, utilisez les commandes **debugreiserfs** et **grep** :

```
root@debian11:~# debugreiserfs /dev/sdc8 | grep UUID
debugreiserfs 3.6.27

UUID: 95c81e34-2215-4b96-be47-e5368444f19a
```

JFS

JFS *Journaled File System* est un système de fichiers journalisé mis au point par IBM et disponible sous licence GPL.

Pour plus d'informations concernant JFS, consultez **cette page**.

Avant de continuer, installer le paquet **jfsutils** :

```
root@debian11:~# apt-get -y install jfsutils
```

LAB #21 - Créer un Système de Fichiers JFS

Créez un système de fichiers JFS sur **/dev/sdc5** :

```
root@debian11:~# mkfs.jfs /dev/sdc5
mkfs.jfs version 1.1.15, 04-Mar-2011
Warning!  All data on device /dev/sdc5 will be lost!

Continue? (Y/N) y
   |

Format completed successfully.

512000 kilobytes total disk space.
```

Consultez maintenant les caractéristiques du système de fichier :

```
root@debian11:~# jfs_tune -l /dev/sdc5
jfs_tune version 1.1.15, 04-Mar-2011

JFS filesystem superblock:

JFS magic number:       'JFS1'
JFS version:            1
JFS state:              clean
JFS flags:              JFS_LINUX  JFS_COMMIT  JFS_GROUPCOMMIT  JFS_INLINELOG
Aggregate block size:   4096 bytes
Aggregate size:         1019464 blocks
Physical block size:    512 bytes
Allocation group size:  8192 aggregate blocks
Log device number:      0x0
Filesystem creation:    Wed Apr 27 13:48:33 2022
Volume label:           ''
```

LAB #15 - Ajouter une Etiquette au Système de Fichiers JFS

Utilisez la commande **jfs_tune** pour associer une étiquette au système de fichiers :

```
root@debian11:~# jfs_tune -L my_jfs /dev/sdc5
jfs_tune version 1.1.15, 04-Mar-2011
Volume label updated successfully.

root@debian11:~# jfs_tune -l /dev/sdc5
jfs_tune version 1.1.15, 04-Mar-2011

JFS filesystem superblock:

JFS magic number:       'JFS1'
JFS version:            1
JFS state:              clean
JFS flags:              JFS_LINUX  JFS_COMMIT  JFS_GROUPCOMMIT  JFS_INLINELOG
Aggregate block size:   4096 bytes
Aggregate size:         1019464 blocks
Physical block size:    512 bytes
Allocation group size:  8192 aggregate blocks
Log device number:      0x0
Filesystem creation:    Wed Apr 27 13:48:33 2022
Volume label:           'my_jfs'
```

Important - Notez que l'étiquette doit être de 16 caractères maximum.

Créez maintenant un UUID pour le système de fichiers :

```
root@debian11:~# jfs_tune -U random /dev/sdc5
```

```
jfs_tune version 1.1.15, 04-Mar-2011
UUID updated successfully.
```

Dernièrement, pour visualiser l'UUID du système de fichiers, utilisez les commandes **jfs_tune** et **grep** :

```
root@debian11:~# jfs_tune -l /dev/sdc5 | grep UUID
File system UUID:       9fc2208f-9b6f-400a-8b0f-4b584f3edaaa
External log UUID:      00000000-0000-0000-d01e-21e2fe7f0000
```

Btrfs

Btrfs, (B-tree file system, prononcé ButterFS) est un système de fichiers expérimental basé sur la copie sur écriture sous licence GNU GPL, développé principalement par Oracle, Red Hat, Fujitsu, Intel, SUSE et STRATO AG, qui s'inspire grandement du système de fichiers ZFS utilisé par Solaris.

A noter sont les points suivants :

- Btrfs utilise des extents,
- Btrfs stocke les données des très petits fichiers directement dans l'extent du fichier répertoire, et non dans un extent séparé,
- Btrfs gère une notion de « sous-volumes » permettant ainsi des snapshots,
- Btrfs possède ses techniques propres de protection des données,
- Btrfs permet de redimensionner à chaud la taille du système de fichiers,
- Btrfs gère le RAID 0 ainsi que le RAID 1 logiciel,
- Btrfs gère la comprssion du système de fichiers.

Avant de continuer, installer le paquet **btrfs-progs** :

```
root@debian11:~# apt-get -y install btrfs-progs
```

LAB #16 - Créer un Système de Fichiers Btrfs

Créez un système de fichiers Btrfs sur **/dev/sdc5** :

```
root@debian11:~# mkfs.btrfs /dev/sdc5
btrfs-progs v5.10.1
See http://btrfs.wiki.kernel.org for more information.

/dev/sdc5 appears to contain an existing filesystem (jfs).
ERROR: use the -f option to force overwrite of /dev/sdc5

root@debian11:~# mkfs.btrfs -f /dev/sdc5
btrfs-progs v5.10.1
See http://btrfs.wiki.kernel.org for more information.

Label:              (null)
UUID:               6030b3a7-a3f0-4af8-b00e-988e43c91784
Node size:          16384
Sector size:        4096
Filesystem size:    500.00MiB
Block group profiles:
  Data:             single            8.00MiB
  Metadata:         DUP              32.00MiB
  System:           DUP               8.00MiB
SSD detected:       no
Incompat features:  extref, skinny-metadata
Runtime features:
Checksum:           crc32c
Number of devices:  1
Devices:
   ID        SIZE  PATH
    1   500.00MiB  /dev/sdc5
```

Montez la partition btrfs sur /mnt :

```
root@debian11:~# mount -t btrfs /dev/sdc5 /mnt/
root@debian11:~# mount | grep btrfs
```

```
/dev/sdc5 on /mnt type btrfs (rw,relatime,space_cache,subvolid=5,subvol=/)
```

Sous Btrfs, il est possible de créer des sous volumes. Pour comprendre, comparez notre partition Btrfs à un VG et des sous volumes comme des LV :

```
root@debian11:~# btrfs subvolume create /mnt/volume1
Create subvolume '/mnt/volume1'

root@debian11:~# btrfs subvolume list /mnt/
ID 256 gen 7 top level 5 path volume1
```

Important - L'ID identifie le volume d'une manière unique.

LAB #17 - Convertir un Système de Fichiers Ext4 en Brtfs

Créez un système de fichiers Ext4 sur **/dev/sdc10** :

```
root@debian11:~# mkfs.ext4 -b 4096 /dev/sdc10
mke2fs 1.46.2 (28-Feb-2021)
/dev/sdc10 contains a ext4 file system
        created on Thu Apr 28 05:36:22 2022
Proceed anyway? (y,N) y
Discarding device blocks: done
Creating filesystem with 128000 4k blocks and 128000 inodes
Filesystem UUID: 592f92f2-9d1b-40bf-89b3-24b35b59621b
Superblock backups stored on blocks:
        32768, 98304

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
```

```
Writing superblocks and filesystem accounting information: done
```

Convertissez maintenant le systèm de fichiers Ext4 en Btrfs :

```
root@debian11:~# btrfs-convert /dev/sdc10
create btrfs filesystem:
        blocksize: 4096
        nodesize:  16384
        features:  extref, skinny-metadata (default)
        checksum:  crc32c
free space report:
        total:     524288000
        free:      390004736 (74.39%)
creating ext2 image file
creating btrfs metadata
copy inodes [o] [         0/        11]
conversion complete
```

Visualisez les systèmes de fichiers Btrfs :

```
root@debian11:~# btrfs filesystem show
Label: none  uuid: 6030b3a7-a3f0-4af8-b00e-988e43c91784
        Total devices 1 FS bytes used 144.00KiB
        devid    1 size 500.00MiB used 88.00MiB path /dev/sdc5

Label: none  uuid: ebdf4e62-3dc5-4bd3-8dc2-988ad5e680e8
        Total devices 1 FS bytes used 32.62MiB
        devid    1 size 500.00MiB used 163.00MiB path /dev/sdc10
```

Ajouter la partition /dev/sdc10 à /mnt :

```
root@debian11:~# btrfs device add /dev/sdc10 /mnt/
/dev/sdc10 appears to contain an existing filesystem (btrfs).
```

```
ERROR: use the -f option to force overwrite of /dev/sdc10

root@debian11:~# btrfs device add -f /dev/sdc10 /mnt/

root@debian11:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G     0  1.9G   0% /dev
tmpfs           394M  904K  393M   1% /run
/dev/sdb1        31G  3.5G   26G  13% /
tmpfs           2.0G     0  2.0G   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           394M   44K  394M   1% /run/user/113
tmpfs           394M   40K  394M   1% /run/user/1000
/dev/sdc5      1000M  3.6M  918M   1% /mnt

root@debian11:~# btrfs filesystem show
Label: none  uuid: 6030b3a7-a3f0-4af8-b00e-988e43c91784
        Total devices 2 FS bytes used 144.00KiB
        devid    1 size 500.00MiB used 88.00MiB path /dev/sdc5
        devid    2 size 500.00MiB used 0.00B path /dev/sdc10

root@debian11:~# btrfs filesystem df /mnt/
Data, single: total=8.00MiB, used=0.00B
System, DUP: total=8.00MiB, used=16.00KiB
Metadata, DUP: total=32.00MiB, used=144.00KiB
GlobalReserve, single: total=3.25MiB, used=0.00B
```

LAB #18 - Créer un Snapshot

Creéz un snapshot du volume /mnt :

```
root@debian11:~# btrfs subvolume snapshot /mnt /mnt/snapshot
Create a snapshot of '/mnt' in '/mnt/snapshot'
```

```
root@debian11:~# ls -l /mnt
total 16
drwxr-xr-x 1 root root 14 Apr 28 05:31 snapshot
drwxr-xr-x 1 root root  0 Apr 28 05:33 volume1
```

Démontez le système de fichiers btrfs :

```
root@debian11:~# umount /mnt
```

Montez ensuite le snapshot :

```
root@debian11:~# mount -o subvol=snapshot /dev/sdc5 /mnt/

root@debian11:~# ls -l /mnt
total 0
drwxr-xr-x 1 root root 0 Apr 28 05:46 volume1

root@debian11:~# btrfs filesystem df /mnt/
Data, single: total=8.00MiB, used=0.00B
System, DUP: total=8.00MiB, used=16.00KiB
Metadata, DUP: total=32.00MiB, used=144.00KiB
GlobalReserve, single: total=3.25MiB, used=0.00B
```

LAB #19 - Ajouter une Etiquette au Système de Fichiers Btrfs

Pour voir les caractéristques du système de fichiers Btrfs, utilisez la commande suivante :

```
root@debian11:~# btrfs inspect-internal dump-super /dev/sdc5
superblock: bytenr=65536, device=/dev/sdc5
---------------------------------------------------------
csum_type               0 (crc32c)
csum_size               4
```

```
csum                    0xc63d381f [match]
bytenr                  65536
flags                   0x1
                        ( WRITTEN )
magic                   _BHRfS_M [match]
fsid                    6030b3a7-a3f0-4af8-b00e-988e43c91784
metadata_uuid           6030b3a7-a3f0-4af8-b00e-988e43c91784
label
generation              12
root                    30441472
sys_array_size          129
chunk_root_generation   8
root_level              0
chunk_root              22020096
chunk_root_level        0
log_root                0
log_root_transid        0
log_root_level          0
total_bytes             1048576000
bytes_used              163840
sectorsize              4096
nodesize                16384
leafsize (deprecated)   16384
stripesize              4096
root_dir                6
num_devices             2
compat_flags            0x0
compat_ro_flags         0x0
incompat_flags          0x141
                        ( MIXED_BACKREF |
                          EXTENDED_IREF |
                          SKINNY_METADATA )
cache_generation        12
uuid_tree_generation    12
```

```
dev_item.uuid           ee13b782-2ecb-443a-8703-b44206bab1e3
dev_item.fsid           6030b3a7-a3f0-4af8-b00e-988e43c91784 [match]
dev_item.type           0
dev_item.total_bytes    524288000
dev_item.bytes_used     92274688
dev_item.io_align       4096
dev_item.io_width       4096
dev_item.sector_size    4096
dev_item.devid          1
dev_item.dev_group      0
dev_item.seek_speed     0
dev_item.bandwidth      0
dev_item.generation     0
```

Créez un une etiquette avec la commande **btrfs filesystem label** :

```
root@debian11:~# btrfs filesystem label /mnt my_btrfs

root@debian11:~# btrfs inspect-internal dump-super /dev/sdc5
superblock: bytenr=65536, device=/dev/sdc5
---------------------------------------------------------
csum_type               0 (crc32c)
csum_size               4
csum                    0x981c9953 [match]
bytenr                  65536
flags                   0x1
                        ( WRITTEN )
magic                   _BHRfS_M [match]
fsid                    6030b3a7-a3f0-4af8-b00e-988e43c91784
metadata_uuid           6030b3a7-a3f0-4af8-b00e-988e43c91784
label                   my_btrfs
generation              13
root                    30523392
sys_array_size          129
```

```
chunk_root_generation   8
root_level              0
chunk_root              22020096
chunk_root_level        0
log_root                0
log_root_transid        0
log_root_level          0
total_bytes             1048576000
bytes_used              163840
sectorsize              4096
nodesize                16384
leafsize (deprecated)   16384
stripesize              4096
root_dir                6
num_devices             2
compat_flags            0x0
compat_ro_flags         0x0
incompat_flags          0x141
                        ( MIXED_BACKREF |
                          EXTENDED_IREF |
                          SKINNY_METADATA )
cache_generation        13
uuid_tree_generation    13
dev_item.uuid           ee13b782-2ecb-443a-8703-b44206bab1e3
dev_item.fsid           6030b3a7-a3f0-4af8-b00e-988e43c91784 [match]
dev_item.type           0
dev_item.total_bytes    524288000
dev_item.bytes_used     92274688
dev_item.io_align       4096
dev_item.io_width       4096
dev_item.sector_size    4096
dev_item.devid          1
dev_item.dev_group      0
dev_item.seek_speed     0
```

```
dev_item.bandwidth      0
dev_item.generation     0
```

Démontez le système de fichiers btrfs :

```
root@debian11:~# umount /mnt
```

Comparaison des Commandes par Système de Fichiers

Description	Ext3	Ext4	XFS	ReiserFS	JFS	Btrfs
Build a Linux filesystem	mkfs.ext3 (mke2fs -j)	mkfs.ext4 (mke4fs)	mkfs.xfs	mkfs.reiserfs (mkreiserfs)	mkfs.jfs (jfs_mkfs)	mkfs.btrfs
Check a Linux filesystem	e2fsck	e2fsck	xfs_check / xfs_repair	reiserfsck	jfs_fsck	btrfsck
Adjust tunable filesystem parameters Linux filesystems	tune2fs	tune2fs	xfs_admin	reiserfstune	jfs_tune	btrfs-show-super, btrfs filesystem show, et btrfs filesystem df
File system resizer	resize2fs	resize2fs	xfs_growfs	resize_reiserfs	S/O	btrfs filesystem resize
Dump filesystem information	dumpe2fs	dumpe2fs	xfs_info / xfs_metadump	debugreiserfs	jfs_tune	btrfstune
File system debugger	debugfs	debugfs	xfs_db	debugreiserfs	jfs_debugfs	btrfs-debug-tree
Change the label on a filesystem	e2label	e2label	xfs_admin	reiserfstune	jfs_tune	btrfs filesystem label

Systèmes de Fichiers Chiffrés

LAB #20 - Créer un Système de Fichiers Chiffré avec encryptfs

Commencez par installer le paquet **ecryptfs-utils** dans la machine virtuelle Debian 11 :

```
root@debian11:~# apt-get -y install ecryptfs-utils
```

Remontez /mnt/sdc12 sur lui-même en spécifiant le type de fichiers en tant qu'encryptfs :

```
root@debian11:~# mount -t ecryptfs /mnt/sdc12 /mnt/sdc12
Select key type to use for newly created files:
 1) tspi
 2) passphrase
Selection: 2
Passphrase:
Select cipher:
 1) aes: blocksize = 16; min keysize = 16; max keysize = 32
 2) blowfish: blocksize = 8; min keysize = 16; max keysize = 56
 3) des3_ede: blocksize = 8; min keysize = 24; max keysize = 24
 4) twofish: blocksize = 16; min keysize = 16; max keysize = 32
 5) cast6: blocksize = 16; min keysize = 16; max keysize = 32
 6) cast5: blocksize = 8; min keysize = 5; max keysize = 16
Selection [aes]: 1
Select key bytes:
 1) 16
 2) 32
 3) 24
Selection [16]: 1
Enable plaintext passthrough (y/n) [n]: n
Enable filename encryption (y/n) [n]: y
Filename Encryption Key (FNEK) Signature [91aefde99b5a4977]:
Attempting to mount with the following options:
  ecryptfs_unlink_sigs
  ecryptfs_fnek_sig=91aefde99b5a4977
  ecryptfs_key_bytes=16
  ecryptfs_cipher=aes
  ecryptfs_sig=91aefde99b5a4977
WARNING: Based on the contents of [/root/.ecryptfs/sig-cache.txt],
it looks like you have never mounted with this key
before. This could mean that you have typed your
passphrase wrong.
```

```
Would you like to proceed with the mount (yes/no)? : yes
Would you like to append sig [91aefde99b5a4977] to
[/root/.ecryptfs/sig-cache.txt]
in order to avoid this warning in the future (yes/no)? : yes
Successfully appended new sig to user sig cache file
Mounted eCryptfs
```

Ce montage est visible dans la sortie de la commande **df** :

```
root@debian11:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G     0  1.9G   0% /dev
tmpfs           394M  904K  393M   1% /run
/dev/sdb1        31G  3.5G   26G  13% /
tmpfs           2.0G     0  2.0G   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           394M   44K  394M   1% /run/user/113
/dev/sdc11      474M   14K  445M   1% /mnt/sdc11
/mnt/sdc12      189M   32K  175M   1% /mnt/sdc12
tmpfs           394M   40K  394M   1% /run/user/1000
```

Plus de détails sont visibles avec la commande **mount** :

```
root@debian11:~# mount | tail
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,pagesize=2M)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime)
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
tmpfs on /run/user/113 type tmpfs
(rw,nosuid,nodev,relatime,size=402556k,nr_inodes=100639,mode=700,uid=113,gid=121)
/dev/sdc11 on /mnt/sdc11 type ext4 (rw,relatime)
/dev/sdc12 on /mnt/sdc12 type ext4 (rw,relatime)
tmpfs on /run/user/1000 type tmpfs
```

```
(rw,nosuid,nodev,relatime,size=402556k,nr_inodes=100639,mode=700,uid=1000,gid=1000)
/mnt/sdc12 on /mnt/sdc12 type ecryptfs
(rw,relatime,ecryptfs_fnek_sig=91aefde99b5a4977,ecryptfs_sig=91aefde99b5a4977,ecryptfs_cipher=aes,ecryptfs_key_by
tes=16,ecryptfs_unlink_sigs)
```

Créez maintenant le fichier **encrypt** contenant la chaîne de caractères **fenestros** dans /mnt/sdc12 :

```
root@debian11:~# touch /mnt/sdc12/encrypt

root@debian11:~# echo "fenestros" > /mnt/sdc12/encrypt

root@debian11:~# cat /mnt/sdc12/encrypt
fenestros
```

Démontez maintenant /mnt/sdc12 :

```
root@debian11:~# umount /mnt/sdc12

root@debian11:~# mount | tail
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,pagesize=2M)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime)
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
tmpfs on /run/user/113 type tmpfs
(rw,nosuid,nodev,relatime,size=402556k,nr_inodes=100639,mode=700,uid=113,gid=121)
/dev/sdc11 on /mnt/sdc11 type ext4 (rw,relatime)
/dev/sdc12 on /mnt/sdc12 type ext4 (rw,relatime)
tmpfs on /run/user/1000 type tmpfs
(rw,nosuid,nodev,relatime,size=402556k,nr_inodes=100639,mode=700,uid=1000,gid=1000)
```

Important : Notez que /dev/sdc12 est toujours monté sur /mnt/sdc12.

Constatez maintenant le contenu de /mnt/sdc12 :

```
root@debian11:~# ls -l /mnt/sdc12
total 25
-rw-r--r-- 1 root root    11 Apr 27 12:15 check_file
-rw-r--r-- 1 root root 12288 Apr 28 06:12
ECRYPTFS_FNEK_ENCRYPTED.FWaFfjrdapd7RkRCT3OFIdaE.-6mxquDYm8R4p7VFuV0SGgSJauuQJ2hEE--
drwx------ 2 root root 12288 Apr 27 12:11 lost+found
```

Important : Notez que le nom du fichier **encrypt** a été chiffré.

Constatez maintenant le contenu du fichier **ECRYPTFS_FNEK_ENCRYPTED.FWaFfjrdapd7RkRCT3OFIdaE.-6mxquDYm8R4p7VFuV0SGgSJauuQJ2hEE-** :

```
root@debian11:~# cat
/mnt/sdc12/ECRYPTFS_FNEK_ENCRYPTED.FWaFfjrdapd7RkRCT3OFIdaE.-6mxquDYm8R4p7VFuV0SGgSJauuQJ2hEE--

KnwM
"3DUfw`6.X0XgRғ2 _CONSOLEZIw#t+X
                                qF3|2m&69CHlú)Tn%A^cư[kL
N                                                          h
5uxΠRNbb&<[RnIpz,0nKjC\y eZ5qmE:K:<\rғϑUY9^E|hW!|0yd\R^B*/h5)Q 8b PjQ]z[l0q
bhXnM`rAgM}{xc\DçYMuŐ=3u|ivcWV3P6<#F#[<TP&ūn;6pF-0!l}uz$zZENCriFWғLMTVj3
?9xdQp$X.x`?L"4;5sQf0|%V5 n"DЋ!*MfsPU9R'(eq^ NU
                        e|\Z._NIBik1>UGD.S'}$9 hdc-`IDX}Qf0\Gv w-Cdfj{+K%m     T=pT)(  H&d`\/yb1}
FБ}ff(JƒĀ,~"-Ou 4
JЮI/2\H[pi
```

```
)B亭v]?~p4=x\OF? *1)
                    v/RD+□{GB8\IGiw□
                                   uAc~anv˛
                     2□|tA=\{5+4_z+oF    L>Xx$*eVCb p(
L&a0Y'$?_P\b;d}7r(`'++bꭗ/0Kmm5-<,3[RkmWGuLOPΦUG_!34$ia* U                   %}(ԁQ2     B*ٻI$.@s?![o-
}S{=,k2k]ĸ8dhn@s%? Tz̄,k]jqc
x;F;$Z0#+0Sd8f\YBc3VĠ蕵IAq
                          >BA4XGlfF$2scB4lo#OZ+yCEZ^_
(@'`'i\B~[[K.ohc6Qw/|                                    y>8ok
                     L
M+LMw   e>發nƅ>rɣΛpB(i□5VY#DA□?z)Òpr(
'3c}%71k !F_Ю`ĪFs!]LD
p/&8`{?H<UI<|1xU2f3F喜1KM𠁥2P)鵮;ي}U?'T9UP@wp娛lkzLǀb<1(ux
*a+n
]HDuՊ:#]`]^Bӹ3LA:euGqHƆ.g致rp湮8N!e)}Gs["aS)bX.Ny;6$□óymMR逓+
                                      ɖpk#D>'
                                                                        %'A<8땅/ph`kF{-Noh+Ç+~;pqoqE]d
)m)'¸t}al*Qh%so?%BBo4OM7&vEBF9i*aApjm8|=QlwmWtz`#ɣ~3(=eR3SfU @u2
n
BD^DoskY}N8S7ۋз^b]1W-?DqQarpv~>!hIkXά@Cow&XJ<Hȩ(uΓu{c/mdˡS&.GFp7euk)OI!-(!%eL0F
x]Ç@0(خ
60嚏      )G*
m^xveʧfo|â'wn^^b[H□_Bg+&;~b     Zvn,Y□4Z2<.}`-sTUn k5(1r\rvf_n{fЩ
                                                                    $f6YWƐ□K4m0|c$!y
                                                                                   /Yܣos3G{zƿ_+~)/z@h1!vn"h
*l+uyBeҳ{p%|V)EauM(N mJ=%XUx-bScF&
6q@dc!B□LG)b6tT6'hD4E8-mӈ
d(<v!RX□ʒd%~IjaЧzb)Q'8j#K9□r+9?,H!q4
:U□X*d
KmPem&btƨesHc^J91Fplq4CL\چF     sIwҕ@*:H& {K.rT7M9<i
q6fuSf93        □a7qP[TrR_V0,Y  n9 i
```

```
                                    pe5
kf,`F'+}VezEuBv,_4 ~g/M90m>ꚩL5,i)c%4~∆a-*au`ꞔ0_~;[9"n]ꞓm#3PrY~(S3:J66fu%DA|wWFI%ꞁr{EȐ(7TK
                                                                                           eI\1Nx^}
G9+|NhKp-hđ-dJh<F                                                                                 .&Ↄaukw6!Q
迸J_FIc|t.],uTlf3vc?4Zy,zBl(]^uZp5"gsBii/:aHak[$kx@6- Ps.@
_4QЛυ$Nn:``ᵽc#M
              73F5
          ɜl:9`Qg䆀3 GwTroot@debian11:~#
```

Important : Notez que le contenu du fichier **encrypt** a été chiffré. Pour pouvoir lire le nom et le contenu de ce fichier de nouveau, il faut remonter /mnt/sdc12 en spécifiant les mêmes options ainsi que la même passphrase. Notez que si vous vous trompez au niveau de la passphrase ceci n'empêchera pas le processus de montage. Par contre vous ne pourrez ni lire le nom ni lire le contenu du fichier chiffré.

LAB #28 - Créer un Système de Fichiers Chiffré avec LUKS

Présentation

LUKS (Linux Unified Key Setup) permet de chiffrer l'intégralité d'un disque de telle sorte que celui-ci soit utilisable sur d'autres plates-formes et distributions de Linux (voire d'autres systèmes d'exploitation). Il supporte des mots de passe multiples, afin que plusieurs utilisateurs soient en mesure de déchiffrer le même volume sans partager leur mot de passe.

Mise en Place

Commencez par installer le paquet **cryptsetup** :

```
root@debian11:/# apt-get -y install cryptsetup
```

Remplissez la partition /dev/sdc11 avec des données aléatoires :

```
root@debian11:~# shred -v --iterations=1 /dev/sdc11
shred: /dev/sdc11: pass 1/1 (random)...
shred: /dev/sdc11: pass 1/1 (random)...474MiB/500MiB 94%
shred: /dev/sdc11: pass 1/1 (random)...500MiB/500MiB 100%
```

Important : L'étape ci-dessus est très importante parce que elle permet de s'assurer qu'aucune donnée ne reste sur la partition.

Démontez /dev/sdc11 :

```
root@debian11:/# umount /mnt/sdc11
```

Initialisez la partition avec LUKS :

```
root@debian11:~# cryptsetup --verbose --verify-passphrase luksFormat /dev/sdc11

WARNING!
========
This will overwrite data on /dev/sdc11 irrevocably.

Are you sure? (Type 'yes' in capital letters): YES
Enter passphrase for /dev/sdc11: fenestros123456789
Verify passphrase: fenestros123456789
Key slot 0 created.
Command successful.
```

Important : La passphrase ne sera pas en claire. Elle l'est ici pour vous montrer un mot de passe acceptable pour LUKS.

Ouvrez la partition LUKS en lui donnant le nom **sdc11** :

```
root@debian11:~# cryptsetup luksOpen /dev/sdc11 sdc11
Enter passphrase for /dev/sdc11: fenestros123456789
```

Vérifiez que le système voit la partition :

```
root@debian11:~# ls -l /dev/mapper | grep sdc11
lrwxrwxrwx 1 root root       7 Apr 28 06:26 sdc11 -> ../dm-2
```

Créez maintenant un système de fichiers sur **/dev/mapper/sdc11** :

```
root@debian11:~# mkfs.ext4 /dev/mapper/sdc11
mke2fs 1.46.2 (28-Feb-2021)
Creating filesystem with 495616 1k blocks and 123952 inodes
Filesystem UUID: cc35e698-6915-4cb3-a0ca-e76bce705be1
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729, 204801, 221185, 401409

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

Montez la partition LUKS :

```
root@debian11:/# mount /dev/mapper/sdc11 /mnt/sdc11
```

Vérifiez la présence du montage :

```
root@debian11:~# df -h | grep sdc11
/dev/mapper/sdc11  459M   14K  431M   1% /mnt/sdc11
```

Editez le fichier **/etc/crypttab/** :

```
root@debian11:~# vi /etc/crypttab

root@debian11:~# cat /etc/crypttab
# <target name> <source device>         <key file>      <options>
sdc11 /dev/sdc11 none
```

Modifiez le fichier **/etc/fstab** :

```
root@debian11:~# vi /etc/fstab

root@debian11:~# 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).
#
# systemd generates mount units based on this file, see systemd.mount(5).
# Please run 'systemctl daemon-reload' after making changes here.
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/sda1 during installation
UUID=9887a74f-a680-4bde-8f04-db5ae9ea186e /               ext4    errors=remount-ro 0       1
# swap was on /dev/sda5 during installation
UUID=1f9439f5-4b19-49b1-b292-60c2c674cee9 none            swap    sw              0       0
/dev/sr0        /media/cdrom0   udf,iso9660 user,noauto     0       0
/dev/mapper/sdc11                                   /mnt/sdc11                      ext4        defaults
1 2
```

Ajouter une deuxième Passphrase

Pour ajouter une deuxième passphrase, utilisez la commande cryptsetup avec la sous-commande **luksAddKey** :

```
root@debian11:~# cryptsetup luksAddKey /dev/sdc11
Enter any existing passphrase: fenestros123456789
Enter new passphrase for key slot: debian123456789
Verify passphrase: debian123456789
```

Important : Les passphrases ne seront pas en claire. Elle le sont ici pour vous montrer des mots de passe acceptables pour LUKS.

Supprimer une Passphrase

Pour supprimer une passphrase, utilisez la commande cryptsetup avec la sous-commande **luksRemoveKey** :

```
root@debian11:~# cryptsetup luksRemoveKey /dev/sdc11
Enter passphrase to be deleted: debian123456789
```

NE PAS FAIRE : Lors du démarrage de la machine virtuelle, le système vous demandera d'entrer la passphrase **fenestros123456789** pour permettre le montage de /dev/sdc11.

Le Swap

Taille du swap

Le tableau suivant résume la taille du swap recommandée en fonction de la mémoire de la machine :

Mémoire	Taille du swap
4 Go ou moins	2 Go
4 Go à 16 Go	4 Go
16 Go à 64 Go	8 Go
64 Go à 256 Go	16 Go

Partitions de swap

Une partition de swap peut être créée sur :

- une partition du disque dur
- un RAID logiciel
- un Volume Logique

La Commande swapon

Pour préparer un espace de swap, il convient d'utiliser la commande **mkswap**. Pour activer une partition de swap, il convient d'utiliser la commande **swapon**. Pour consulter la liste des partitions swap, il convient d'utiliser la commande **swapon** avec l'option **-s**.

```
[root@centos8 ~]# swapon -s
Filename                Type        Size    Used    Priority
/dev/dm-1                               partition       3358716 0       -2
```

Important : Vous noterez que dans l'exemple ci-dessus, le swap n'est pas utilisé. Notez aussi qu'il existe une notion de **priorité** pour les partions de swap.

Options de la Commande

Les options de la commande swapon sont :

```
[root@centos8 ~]# swapon --help

Usage:
 swapon [options] [<spec>]

Enable devices and files for paging and swapping.

Options:
 -a, --all                enable all swaps from /etc/fstab
 -d, --discard[=<policy>] enable swap discards, if supported by device
 -e, --ifexists           silently skip devices that do not exist
 -f, --fixpgsz            reinitialize the swap space if necessary
 -o, --options <list>     comma-separated list of swap options
 -p, --priority <prio>    specify the priority of the swap device
 -s, --summary            display summary about used swap devices (DEPRECATED)
     --show[=<columns>]   display summary in definable table
     --noheadings         don't print table heading (with --show)
     --raw                use the raw output format (with --show)
     --bytes              display swap size in bytes in --show output
 -v, --verbose            verbose mode

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

The <spec> parameter:
 -L <label>             synonym for LABEL=<label>
 -U <uuid>              synonym for UUID=<uuid>
 LABEL=<label>          specifies device by swap area label
 UUID=<uuid>            specifies device by swap area UUID
```

```
 PARTLABEL=<label>      specifies device by partition label
 PARTUUID=<uuid>        specifies device by partition UUID
 <device>               name of device to be used
 <file>                 name of file to be used

Available discard policy types (for --discard):
 once    : only single-time area discards are issued
 pages   : freed pages are discarded before they are reused
If no policy is selected, both discard types are enabled (default).

Available output columns:
 NAME   device file or partition path
 TYPE   type of the device
 SIZE   size of the swap area
 USED   bytes in use
 PRIO   swap priority
 UUID   swap uuid
 LABEL  swap label

For more details see swapon(8).
```

Important : L'option **-p** de la commande **swapon** permet de régler la priorité.

La Commande swapoff

Dans le cas de notre exemple, la partition de swap se trouve sur **/dev/dm-1**. Pour la désactiver, il convient de saisir la commande suivante :

```
[root@centos8 ~]# swapoff /dev/dm-1
[root@centos8 ~]# swapon -s
```

```
[root@centos8 ~]#
```

Options de la Commande

```
[root@centos8 ~]# swapoff --help

Usage:
 swapoff [options] [<spec>]

Disable devices and files for paging and swapping.

Options:
 -a, --all              disable all swaps from /proc/swaps
 -v, --verbose          verbose mode

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

The <spec> parameter:
 -L <label>             LABEL of device to be used
 -U <uuid>              UUID of device to be used
 LABEL=<label>          LABEL of device to be used
 UUID=<uuid>            UUID of device to be used
 <device>               name of device to be used
 <file>                 name of file to be used

For more details see swapoff(8).
```

LAB #13 - Créer un Fichier de Swap

Sous Linux, vous pouvez aussi bien utiliser un fichier de swap qu'une partition. La mise en place de ce fichier est faite en utilisant la commande **dd**.

La commande **dd** copie le fichier passé en entrée dans le fichier de sortie en limitant le nombre d'octets copiés par l'utilisation de deux options :

- **count**
  - le nombre
- **bs**
  - la taille du bloc à copier

Dans le cas du fichier swap il convient d'utiliser le fichier spécial **/dev/zero** en tant que fichier d'entrée. Le fichier **/dev/zero** contient une valeur **null**.

Pour créer votre fichier de swap de 268Mo, appelé **swap**, saisissez la commande suivante :

```
[root@centos8 ~]# dd if=/dev/zero of=/swap bs=1024k count=256
256+0 records in
256+0 records out
268435456 bytes (268 MB, 256 MiB) copied, 0.103048 s, 2.6 GB/s
```

Pour préparer le fichier en tant qu'espace de swap, saisissez la commande suivante :

```
[root@centos8 ~]# mkswap /swap
mkswap: /swap: insecure permissions 0644, 0600 suggested.
Setting up swapspace version 1, size = 256 MiB (268431360 bytes)
no label, UUID=ee6c9e3f-0712-47b1-8f97-17ba215959d7
```

Pour activer le fichier avec une priorité de **1**, saisissez la commande suivante :

```
[root@centos8 ~]# swapon -p1 /swap
swapon: /swap: insecure permissions 0644, 0600 suggested.
```

Pour visualiser les éspaces swap, saisissez la commande suivante :

```
[root@centos8 ~]# swapon -s
Filename                Type        Size    Used    Priority
/swap                                   file        262140    0    1
/dev/dm-1                               partition       3358716 0       -2
```

Important : Le fichier de swap ayant une priorité de 1 sera utilisé avant la partition de swap ayant une priorité de -2.

Important : Pour activer le fichier swap d'une manière permanente, il convient d'ajouter une ligne au fichier **/etc/fstab**. Ne modifiez pas votre fichier **/etc/fstab** car vous allez supprimer le fichier de swap.

Désactivez maintenant le fichier swap :

```
[root@centos8 ~]# swapoff /swap
[root@centos8 ~]# swapon -s
Filename                Type        Size    Used    Priority
/dev/dm-1                               partition       3358716 0       -2
```

Supprimez maintenant le fichier de swap :

```
[root@centos8 ~]# rm /swap
rm: remove regular file '/swap'? y
```

Commandes Diverses

La Commande sync

La commande sync synchronise les données de fichier correspondantes dans la mémoire volatile et le stockage permanent. Les écritures en cache sont immédiatement écrites sur le disque.

Les options de cette commande sont :

```
[root@centos8 ~]# sync --help
Usage: sync [OPTION] [FILE]...
Synchronize cached writes to persistent storage

If one or more files are specified, sync only them,
or their containing file systems.

  -d, --data             sync only file data, no unneeded metadata
  -f, --file-system      sync the file systems that contain the files
      --help     display this help and exit
      --version  output version information and exit

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation at: <https://www.gnu.org/software/coreutils/sync>
or available locally via: info '(coreutils) sync invocation'
```

La Commande fstrim

La commande **fstrim** est utilisée sur un système de fichiers monté pour abandonner (ou rogner) les blocs qui ne sont pas utilisés par le système de fichiers. C'est pratique pour les SSD et l'allocation fine et dynamique (thinly-provisioned storage).

Les options de cette commande sont :

```
[root@centos8 ~]# fstrim --help

Usage:
 fstrim [options] <mount point>

Discard unused blocks on a mounted filesystem.

Options:
```

```
 -a, --all           trim all mounted filesystems that are supported
 -o, --offset <num>  the offset in bytes to start discarding from
 -l, --length <num>  the number of bytes to discard
 -m, --minimum <num> the minimum extent length to discard
 -v, --verbose       print number of discarded bytes

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

For more details see fstrim(8).
```

Le Daemon smartd

SMART (Self-Monitoring, Analysis and Reporting Technology) est intégrée à la plupart des disques durs pour avoir des indicateurs de diagnostic. **Smartmontools** est l'outil pour exploiter la technologie SMART, notamment avec la commande smartctl et le démon smartd.
