

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

LSF109 - Gestion des Disques, des Systèmes de Fichiers et du Swap



Important : Arrêtez votre machine virtuelle. Ajoutez un deuxième disque dur de type vmdk de nommé **SLES_12SP1-disk2.vmdk** au contrôleur SATA. Démarrez votre machine virtuelle.

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 accéder 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

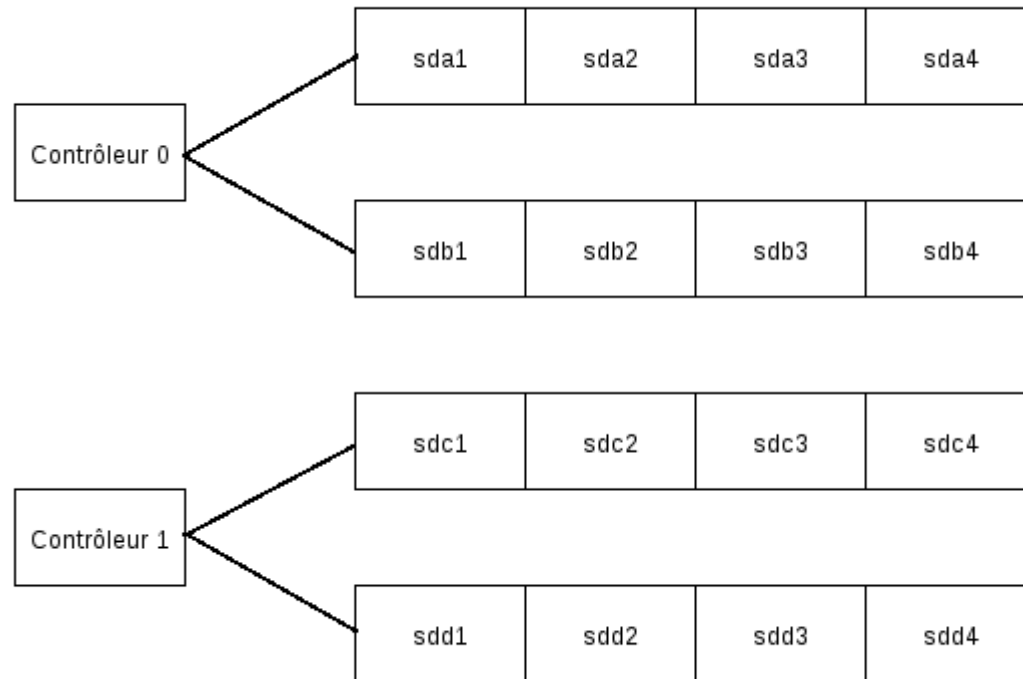
- Maître
 - **hdc** - disque IDE
 - **sdc** - disque SATA ou SCSI
- Esclave
 - **hdd** - disque IDE
 - **sdd** - disque SATA ou SCSI

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 coder une partition.
- **Partitions Étendues,**
 - Généralement une seule partition étendue par disque. Elle contient des **Lecteurs Logiques** aussi appelés des partitions,
- **Lecteurs Logiques.**

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, SLES possède l'outil dénommé **fdisk**.

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

```
SUSE12SP1:~ # fdisk /dev/sdb
```

```
Welcome to fdisk (util-linux 2.25).
```

```
Changes will remain in memory only, until you decide to write them.
```

Be careful before using the write command.

Command (m for help):

Tapez ensuite la lettre **m** puis 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
- 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

Misc

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

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/sdb1	Primary	500 Mo
/dev/sdb2	Primary	200 Mo
/dev/sdb3	Primary	300 Mo
/dev/sdb4	Extended	Du premier cylindre disponible au dernier cylindre du disque
/dev/sdb5	Logical	500 Mo
/dev/sdb6	Logical	400 Mo
/dev/sdb7	Logical	500 Mo
/dev/sdb8	Logical	500 Mo
/dev/sdb9	Logical	200 Mo

Command (m for help): n

Partition type

```

p  primary (0 primary, 0 extended, 4 free)
e  extended (container for logical partitions)

```

Select (default p): p

Partition number (1-4, default 1): 1

First sector (2048-41943039, default 2048):

Last sector, +sectors or +size{K,M,G,T,P} (2048-41943039, default 41943039): +500M

Created a new partition 1 of type 'Linux' and of size 500 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): p
Partition number (2-4, default 2): 2
First sector (1026048-41943039, default 1026048):
Last sector, +sectors or +size{K,M,G,T,P} (1026048-41943039, default 41943039): +200M
```

Created a new partition 2 of type 'Linux' and of size 200 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): p
Partition number (3,4, default 3): 3
First sector (1435648-41943039, default 1435648):
Last sector, +sectors or +size{K,M,G,T,P} (1435648-41943039, default 41943039): +300M
```

Created a new partition 3 of type 'Linux' and of size 300 MiB.

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

Selected partition 4
First sector (2050048-41943039, default 2050048):
Last sector, +sectors or +size{K,M,G,T,P} (2050048-41943039, default 41943039):
```

Created a new partition 4 of type 'Extended' and of size 19 GiB.


```
Command (m for help): n
All primary partitions are in use.
Adding logical partition 5
First sector (2052096-41943039, default 2052096):
Last sector, +sectors or +size{K,M,G,T,P} (2052096-41943039, default 41943039): +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 (3078144-41943039, default 3078144):
Last sector, +sectors or +size{K,M,G,T,P} (3078144-41943039, default 41943039): +400M

Created a new partition 6 of type 'Linux' and of size 400 MiB.

Command (m for help): n
All primary partitions are in use.
Adding logical partition 7
First sector (3899392-41943039, default 3899392):
Last sector, +sectors or +size{K,M,G,T,P} (3899392-41943039, default 41943039): +500M

Created a new partition 7 of type 'Linux' and of size 500 MiB.

Command (m for help): n
All primary partitions are in use.
Adding logical partition 8
First sector (4925440-41943039, default 4925440):
Last sector, +sectors or +size{K,M,G,T,P} (4925440-41943039, default 41943039): +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 (5951488-41943039, default 5951488):
Last sector, +sectors or +size{K,M,G,T,P} (5951488-41943039, default 41943039): +200M

Created a new partition 9 of type 'Linux' and of size 200 MiB.

Command (m for help):
```

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

```
Command (m for help): p
Disk /dev/sdb: 20 GiB, 21474836480 bytes, 41943040 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
Disklabel type: dos
Disk identifier: 0x07e7d102

Device      Boot  Start      End  Sectors  Size Id Type
/dev/sdb1           2048  1026047  1024000  500M 83 Linux
/dev/sdb2       1026048  1435647   409600  200M 83 Linux
/dev/sdb3       1435648  2050047   614400  300M 83 Linux
/dev/sdb4       2050048 41943039 39892992   19G  5 Extended
/dev/sdb5       2052096  3076095  1024000  500M 83 Linux
/dev/sdb6       3078144  3897343   819200  400M 83 Linux
/dev/sdb7       3899392  4923391  1024000  500M 83 Linux
/dev/sdb8       4925440  5949439  1024000  500M 83 Linux
/dev/sdb9       5951488  6361087   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.
```

```
SUSE12SP1:~ #
```

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

```
SUSE12SP1:~ # fdisk /dev/sdb
```

```
Welcome to fdisk (util-linux 2.25).  
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/sdb: 20 GiB, 21474836480 bytes, 41943040 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  
Disklabel type: dos  
Disk identifier: 0x07e7d102
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	1026047	1024000	500M	83	Linux
/dev/sdb2		1026048	1435647	409600	200M	83	Linux
/dev/sdb3		1435648	2050047	614400	300M	83	Linux
/dev/sdb4		2050048	41943039	39892992	19G	5	Extended
/dev/sdb5		2052096	3076095	1024000	500M	83	Linux
/dev/sdb6		3078144	3897343	819200	400M	83	Linux
/dev/sdb7		3899392	4923391	1024000	500M	83	Linux
/dev/sdb8		4925440	5949439	1024000	500M	83	Linux
/dev/sdb9		5951488	6361087	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 (1-9, default 9): 9

Partition 9 has been deleted.

Command (m for help): p
Disk /dev/sdb: 20 GiB, 21474836480 bytes, 41943040 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
Disklabel type: dos
Disk identifier: 0x07e7d102
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	1026047	1024000	500M	83	Linux
/dev/sdb2		1026048	1435647	409600	200M	83	Linux
/dev/sdb3		1435648	2050047	614400	300M	83	Linux
/dev/sdb4		2050048	41943039	39892992	19G	5	Extended
/dev/sdb5		2052096	3076095	1024000	500M	83	Linux
/dev/sdb6		3078144	3897343	819200	400M	83	Linux
/dev/sdb7		3899392	4923391	1024000	500M	83	Linux
/dev/sdb8		4925440	5949439	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
```

```
SUSE12SP1:~ # fdisk /dev/sdb
```

```
Welcome to fdisk (util-linux 2.25).
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/sdb: 20 GiB, 21474836480 bytes, 41943040 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
Disklabel type: dos
Disk identifier: 0x07e7d102
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	1026047	1024000	500M	83	Linux
/dev/sdb2		1026048	1435647	409600	200M	83	Linux
/dev/sdb3		1435648	2050047	614400	300M	83	Linux
/dev/sdb4		2050048	41943039	39892992	19G	5	Extended
/dev/sdb5		2052096	3076095	1024000	500M	83	Linux
/dev/sdb6		3078144	3897343	819200	400M	83	Linux
/dev/sdb7		3899392	4923391	1024000	500M	83	Linux
/dev/sdb8		4925440	5949439	1024000	500M	83	Linux
/dev/sdb9		5951488	6361087	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	Drapeau
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 (1-9, default 9): 1
Hex code (type L to list all codes): fd

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

Command (m for help): t
Partition number (1-9, default 9): 2
Hex code (type L to list all codes): 8e

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

Command (m for help): t
Partition number (1-9, default 9): 3
Hex code (type L to list all codes): 8e

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

Command (m for help): t
Partition number (1-9, default 9): 5
Hex code (type L to list all codes): fd
```

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

```
Command (m for help): t  
Partition number (1-9, default 9): 6  
Hex code (type L to list all codes): 8e
```

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

```
Command (m for help): t  
Partition number (1-9, default 9): 7  
Hex code (type L to list all codes): fd
```

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

```
Command (m for help): t  
Partition number (1-9, default 9): 8  
Hex code (type L to list all codes): 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/sdb: 20 GiB, 21474836480 bytes, 41943040 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  
Disklabel type: dos  
Disk identifier: 0x07e7d102
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	1026047	1024000	500M	fd	Linux raid autodetect

```
/dev/sdb2      1026048  1435647  409600  200M 8e Linux LVM
/dev/sdb3      1435648  2050047  614400  300M 8e Linux LVM
/dev/sdb4      2050048  41943039 39892992  19G  5 Extended
/dev/sdb5      2052096  3076095  1024000  500M fd Linux raid autodetect
/dev/sdb6      3078144  3897343  819200  400M 8e Linux LVM
/dev/sdb7      3899392  4923391  1024000  500M fd Linux raid autodetect
/dev/sdb8      4925440  5949439  1024000  500M fd Linux raid autodetect
/dev/sdb9      5951488  6361087  409600  200M 83 Linux
```

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

Pour écrire la nouvelle table des partitions sur disque, vous devez utiliser 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.
```

```
SUSE12SP1:~ #
```

Options de la Commande fdisk

Les options de cette commande sont :

```
SUSE12SP1:~ # fdisk --help
```

Usage:

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

Options:

```
-b, --sector-size <size>   physical and logical sector size
```



```
-c, --compatibility[=<mode>] mode is 'dos' or 'nondos' (default)
-L, --color[=<when>]         colorize output (auto, always or never)
-l, --list                   display partitions end exit
-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]

-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 and exit
-V, --version                 output version information and exit
```

For more details see `fdisk(8)`.

Logical Volume Manager (LVM)

LAB #3 - Volumes Logiques Linéaires

Afin de mettre en place le LVM, vous avez besoin du paquet **lvm2** et du paquet **device-mapper**.

Nous allons travailler avec les partitions suivantes :

/dev/sdb2	1026048	1435647	409600	200M	8e	Linux	LVM
/dev/sdb3	1435648	2050047	614400	300M	8e	Linux	LVM
/dev/sdb6	3078144	3897343	819200	400M	8e	Linux	LVM

Pour initialiser le LVM saisissez la commande suivante :

```
SUSE12SP1:~ # vgscan
```

```
Reading all physical volumes. This may take a while...
```

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

```
SUSE12SP1:~ # vgscan --help
vgscan: Search for all volume groups

vgscan [--cache]
        [--commandprofile ProfileName]
        [-d|--debug]
        [-h|--help]
        [--ignorelockingfailure]
        [--mknodes]
        [-P|--partial]
        [-v|--verbose]
        [--version]
```

Physical Volume (PV)

Pour créer le **PV** il convient d'utiliser la commande **pvcreate** :

```
SUSE12SP1:~ # pvcreate /dev/sdb2 /dev/sdb3 /dev/sdb6
Physical volume "/dev/sdb2" successfully created
Physical volume "/dev/sdb3" successfully created
Physical volume "/dev/sdb6" successfully created
```

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

```
SUSE12SP1:~ # pvcreate --help
pvcreate: Initialize physical volume(s) for use by LVM

pvcreate
```

```
[--norestorefile]
[--restorefile file]
[--commandprofile ProfileName]
[-d|--debug]
[-f[f]|--force [--force]]
[-h|-?|--help]
[--labelsector sector]
[-M|--metadatatype 1|2]
[--pvmetadatacopies #copies]
[--bootloaderareaseize BootLoaderAreaSize[bBsSkKmMgGtTpPeE]]
[--metadatasize MetadataSize[bBsSkKmMgGtTpPeE]]
[--dataalignment Alignment[bBsSkKmMgGtTpPeE]]
[--dataalignmentoffset AlignmentOffset[bBsSkKmMgGtTpPeE]]
[--setphysicalvolumesize PhysicalVolumeSize[bBsSkKmMgGtTpPeE]]
[-t|--test]
[-u|--uuid uuid]
[-v|--verbose]
[-y|--yes]
[-Z|--zero {y|n}]
[--version]
PhysicalVolume [PhysicalVolume...]
```

Pour visualiser le PV il convient d'utiliser la commande **pvdisplay** :

```
SUSE12SP1:~ # pvdisplay /dev/sdb2 /dev/sdb3 /dev/sdb6
"/dev/sdb3" is a new physical volume of "300.00 MiB"
--- NEW Physical volume ---
PV Name           /dev/sdb3
VG Name
PV Size           300.00 MiB
Allocatable       NO
PE Size           0
Total PE          0
Free PE           0
```

```
Allocated PE          0
PV UUID               lnUsag-cM0o-76LP-PffQ-wJQK-0Nka-vjtFzy
"/dev/sdb6" is a new physical volume of "400.00 MiB"
--- NEW Physical volume ---
PV Name               /dev/sdb6
VG Name
PV Size               400.00 MiB
Allocatable           NO
PE Size               0
Total PE              0
Free PE               0
Allocated PE          0
PV UUID               f006lr-02qe-ccmx-4JV6-98zm-CGBr-EWQUho
"/dev/sdb2" is a new physical volume of "200.00 MiB"
--- NEW Physical volume ---
PV Name               /dev/sdb2
VG Name
PV Size               200.00 MiB
Allocatable           NO
PE Size               0
Total PE              0
Free PE               0
Allocated PE          0
PV UUID               SdDEnc-2fiL-IY82-K0zp-MScx-w1Nj-zefWFA
```

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

```
SUSE12SP1:~ # pvdisplay --help
pvdisplay: Display various attributes of physical volume(s)

pvdisplay
  [-c|--colon]
  [--commandprofile ProfileName]
  [-d|--debug]
```

```
[--foreign]
[-h|--help]
[--ignorelockingfailure]
[--ignorestalecluster]
[-m|--maps]
[--nosuffix]
[--readonly]
[-S|--select Selection]
[-s|--short]
[--units hHbBsSkKmMgGtTpPeE]
[-v|--verbose]
[--version]
[PhysicalVolumePath [PhysicalVolumePath...]]
```

```
pvdiskdisplay --columns|-C
```

```
[--aligned]
[-a|--all]
[--binary]
[--commandprofile ProfileName]
[-d|--debug]
[--foreign]
[-h|--help]
[--ignorelockingfailure]
[--ignorestalecluster]
[--noheadings]
[--nosuffix]
[-o|--options [+]Field[,Field]]
[-O|--sort [+]-key1[,+|-key2[,...]]]
[-S|--select Selection]
[--readonly]
[--separator Separator]
[--unbuffered]
[--units hHbBsSkKmMgGtTpPeE]
[-v|--verbose]
```

```
[--version]  
[PhysicalVolumePath [PhysicalVolumePath...]]
```

Volume Group (VG) et Physical Extent (PE)

Pour créer un Volume Group dénommé **vg0**, il convient d'utiliser la commande **vgcreate** :

```
SUSE12SP1:~ # vgcreate -s 8M vg0 /dev/sdb2 /dev/sdb3 /dev/sdb6  
Volume group "vg0" successfully created
```

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

```
SUSE12SP1:~ # vgcreate --help  
vgcreate: Create a volume group  
  
vgcreate  
[-A|--autobackup {y|n}]  
[--addtag Tag]  
[--alloc AllocationPolicy]  
[-c|--clustered {y|n}]  
[--commandprofile ProfileName]  
[-d|--debug]  
[-h|--help]  
[-l|--maxlogicalvolumes MaxLogicalVolumes]  
[--metadataprofile ProfileName]  
[-M|--metadatatype 1|2]  
[--[vg]metadaticopies #copies]  
[-p|--maxphysicalvolumes MaxPhysicalVolumes]  
[-s|--physicalextentsize PhysicalExtentSize[bBsSkKmMgGtTpPeE]]  
[--systemid SystemID]  
[-t|--test]  
[-v|--verbose]
```

```
[--version]
[-y|--yes]
[ PHYSICAL DEVICE OPTIONS ]
VolumeGroupName PhysicalDevicePath [PhysicalDevicePath...]
```

Pour afficher les informations concernant **vg0**, il convient d'utiliser la commande **vgdisplay** :

```
SUSE12SP1:~ # vgdisplay vg0
--- Volume group ---
VG Name                vg0
System ID
Format                 lvm2
Metadata Areas        3
Metadata Sequence No  1
VG Access              read/write
VG Status              resizable
MAX LV                 0
Cur LV                0
Open LV               0
Max PV                 0
Cur PV                3
Act PV                3
VG Size                880.00 MiB
PE Size                8.00 MiB
Total PE              110
Alloc PE / Size       0 / 0
Free PE / Size        110 / 880.00 MiB
VG UUID                N1cq1a-DCCc-2zsF-Xjn7-vqPJ-XmIO-1wI6p8
```

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

```
SUSE12SP1:~ # vgdisplay --help
vgdisplay: Display volume group information
```

vgdisplay

```
[-A|--activevolumegroups]
[-c|--colon | -s|--short | -v|--verbose]
[--commandprofile ProfileName]
[-d|--debug]
[--foreign]
[-h|--help]
[--ignorelockingfailure]
[--ignorestalecluster]
[--nosuffix]
[-P|--partial]
[--readonly]
[-S|--select Selection]
[--units hHbBsSkKmGtTpPeE]
[--version]
[VolumeGroupName [VolumeGroupName...]]
```

vgdisplay --columns|-C

```
[--aligned]
[--binary]
[--commandprofile ProfileName]
[-d|--debug]
[--foreign]
[-h|--help]
[--ignorelockingfailure]
[--ignorestalecluster]
[--noheadings]
[--nosuffix]
[-o|--options [+]Field[,Field]]
[-O|--sort [+]-key1[,[+]-key2[,...]]]
[-P|--partial]
[-S|--select Selection]
[--readonly]
[--separator Separator]
```



```
[--unbuffered]
[--units hHbBsSkKmMgGtTpPeE]
[--verbose]
[--version]
[VolumeGroupName [VolumeGroupName...]]
```

Logical Volumes (LV)

Pour créer un **Logical Volume** dénommé **lv0** dans le **Volume Group vg0**, il convient d'utiliser la commande **lvcreate** :

```
SUSE12SP1:~ # lvcreate -L 350 -n lv0 vg0
Rounding up size to full physical extent 352.00 MiB
Logical volume "lv0" created.
```



Important - Notez que la taille du LV est un multiple du PE.

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

```
SUSE12SP1:~ # lvcreate --help
lvcreate: Create a logical volume

lvcreate
  [-A|--autobackup {y|n}]
  [-a|--activate [a|e|l]{y|n}]
  [--addtag Tag]
  [--alloc AllocationPolicy]
  [-H|--cache
    [--cachemode {writeback|writethrough}]]
  [--cachepool CachePoolLogicalVolume{Name|Path}]
  [-c|--chunksize ChunkSize]
```

```
[-C|--contiguous {y|n}]
[--commandprofile ProfileName]
[-d|--debug]
[-h|-?|--help]
[--errorwhenfull {y|n}]
[--ignoremonitoring]
[--monitor {y|n}]
[-i|--stripes Stripes [-I|--stripesize StripeSize]]
[-k|--setactivationskip {y|n}]
[-K|--ignoreactivationskip]
{-l|--extents LogicalExtentsNumber[%{VG|PVS|FREE}] |
-L|--size LogicalVolumeSize[bBsSkKmMgGtTpPeE]}
[-M|--persistent {y|n}] [-j|--major major] [--minor minor]
[--metadataprofile ProfileName]
[-m|--mirrors Mirrors [--nosync]
  [--mirrorlog {disk|core|mirrored}|--corelog]]]
[-n|--name LogicalVolumeName]
[--noudevsync]
[-p|--permission {r|rw}]
[--poolmetadatasize MetadataSize[bBsSkKmMgG]]
[--poolmetadataspare {y|n}]
[--[raid]minrecoveryrate Rate]
[--[raid]maxrecoveryrate Rate]
[-r|--readahead {ReadAheadSectors|auto|none}]
[-R|--regionsize MirrorLogRegionSize]
[-T|--thin
  [--discards {ignore|nopassdown|passdown}]]
[--thinpool ThinPoolLogicalVolume{Name|Path}]
[-t|--test]
[--type VolumeType]
[-v|--verbose]
[-W|--wipesignatures {y|n}]
[-Z|--zero {y|n}]
[--version]
```

```
VolumeGroupName [PhysicalVolumePath...]
```

```
lvcreate
```

```
{ {-s|--snapshot} OriginalLogicalVolume[Path] |  
  [-s|--snapshot] VolumeGroupName[Path] -V|--virtualsize VirtualSize}  
{-H|--cache} VolumeGroupName[Path][OriginalLogicalVolume]  
{-T|--thin} VolumeGroupName[Path][PoolLogicalVolume]  
  -V|--virtualsize VirtualSize}  
[-A|--autobackup {y|n}]  
[--addtag Tag]  
[--alloc AllocationPolicy]  
[--cachepolicy Policy] [--cachesettings Key=Value]  
[-c|--chunksize]  
[-C|--contiguous {y|n}]  
[--commandprofile ProfileName]  
[-d|--debug]  
[--discards {ignore|nopassdown|passdown}]  
[-h|-?|--help]  
[--ignoremonitoring]  
[--monitor {y|n}]  
[-i|--stripes Stripes [-I|--stripesize StripeSize]]  
[-k|--setactivationskip {y|n}]  
[-K|--ignoreactivationskip]  
{-l|--extents LogicalExtentsNumber[%{VG|FREE|ORIGIN}]} |  
  -L|--size LogicalVolumeSize[bBsSkKmMgGtTpPeE]}  
[--poolmetadatasize MetadataVolumeSize[bBsSkKmMgG]]  
[-M|--persistent {y|n}] [-j|--major major] [--minor minor]  
[--metadataprofile ProfileName]  
[-n|--name LogicalVolumeName]  
[--noudevsync]  
[-p|--permission {r|rw}]  
[-r|--readahead ReadAheadSectors|auto|none]  
[-t|--test]  
{--thinpool ThinPoolLogicalVolume[Path] |
```

```
--cachepool CachePoolLogicalVolume[Path]]  
[-v|--verbose]  
[--version]  
[PhysicalVolumePath...]
```

Créez maintenant un répertoire dans /mnt pour monter lv0 :

```
SUSE12SP1:~ # mkdir /mnt/lvm
```

Créez un système de fichiers en **ext3** sur /dev/vg0/lv0 :

```
SUSE12SP1:~ # mke2fs -j /dev/vg0/lv0  
mke2fs 1.42.11 (09-Jul-2014)  
Creating filesystem with 360448 1k blocks and 90112 inodes  
Filesystem UUID: 07f7376a-11b8-46dc-bd1d-f6a7a5aa3e11  
Superblock backups stored on blocks:  
    8193, 24577, 40961, 57345, 73729, 204801, 221185  
  
Allocating group tables: done  
Writing inode tables: done  
Creating journal (8192 blocks): done  
Writing superblocks and filesystem accounting information: done
```

Montez votre lv0 :

```
SUSE12SP1:~ # mount -t ext3 /dev/vg0/lv0 /mnt/lvm
```

Vous allez maintenant copier le contenu de votre répertoire /home vers /mnt/lvm.

Saisissez donc la commande pour copier le contenu de /home :

```
SUSE12SP1:~ # cp -a /home /mnt/lvm
```

Constatez ensuite le contenu de /mnt/lvm :

```
SUSE12SP1:~ # ls -l /mnt/lvm
total 13
drwxr-xr-x 3 root root 1024 Oct  9 11:57 home
drwx----- 2 root root 12288 Oct 10 16:38 lost+found
```

Une particularité du volume logique est la capacité de d'être agrandi ou réduit sans pertes de données. Commencez par constater la taille totale du volume :

```
SUSE12SP1:~ # df -h /mnt/lvm
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/vg0-lv0 333M  2.6M  313M   1% /mnt/lvm
```

Dans la cas de notre exemple, la taille est de **333 Mo** avec **2,6 Mo** occupé.

LAB #4 - Etendre un Volume Logique à Chaud

Pour agrandir un volume logique, le paquet **lvm2** contient les commandes **lvextend** et **resize2fs** :

```
SUSE12SP1:~ # lvextend -L +100M /dev/vg0/lv0
Rounding size to boundary between physical extents: 104.00 MiB
Size of logical volume vg0/lv0 changed from 352.00 MiB (44 extents) to 456.00 MiB (57 extents).
Logical volume lv0 successfully resized
```



Important - Notez que l'agrandissement du volume est un multiple du PE.

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

```
SUSE12SP1:~ # lvextend --help
lvextend: Add space to a logical volume
```

```
lvextend
  [-A|--autobackup y|n]
  [--alloc AllocationPolicy]
  [--commandprofile ProfileName]
  [-d|--debug]
  [-f|--force]
  [-h|--help]
  [-i|--stripes Stripes [-I|--stripesize StripeSize]]
  {-l|--extents [+]LogicalExtentsNumber[%{VG|LV|PVS|FREE|ORIGIN}]} |
  -L|--size [+]LogicalVolumeSize[bBsSkKmMgGtTpPeE]}
  --poolmetadatasize [+]MetadataVolumeSize[bBsSkKmMgG]}
  [-m|--mirrors Mirrors]
  [--nosync]
  [--use-policies]
  [-n|--nofsck]
  [--noudevsync]
  [-r|--resizefs]
  [-t|--test]
  [--type VolumeType]
  [-v|--verbose]
  [--version]
  LogicalVolume[Path] [ PhysicalVolumePath... ]
```

Le volume ayant été agrandi, il est nécessaire maintenant d'agrandir le filesystem qui s'y trouve :

```
SUSE12SP1:~ # resize2fs /dev/vg0/lv0
resize2fs 1.42.11 (09-Jul-2014)
Filesystem at /dev/vg0/lv0 is mounted on /mnt/lvm; on-line resizing required
old_desc_blocks = 2, new_desc_blocks = 2
The filesystem on /dev/vg0/lv0 is now 466944 blocks long.
```

Constatez maintenant la modification de la taille du volume :

```
SUSE12SP1:~ # df -h /mnt/lvm
```

```
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/vg0-lv0 434M  2.8M  410M   1% /mnt/lvm
```

Vous noterez que la taille a augmentée mais que les données sont toujours présentes.

LAB #5 - Snapshots

Un snapshot est un instantané d'un système de fichiers. Dans cet exemple, vous allez créer un snapshot de votre lv0 :

Avant de commencer, créez un fichier de 10Mo dans le volume :

```
SUSE12SP1:~ # dd if=/dev/zero of=/mnt/lvm/10M bs=1048576 count=10
10+0 records in
10+0 records out
10485760 bytes (10 MB) copied, 0.101338 s, 103 MB/s
```

Créez maintenant un snapshot :

```
SUSE12SP1:~ # lvcreate -s -L 15M -n testsnap /dev/vg0/lv0
Rounding up size to full physical extent 16.00 MiB
Logical volume "testsnap" created.
```

Pour avoir une confirmation de la création du snapshot, utilisez la commande **lvs** :

```
SUSE12SP1:~ # lvs
LV      VG   Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
lv0     vg0  owi-aos--- 456.00m
testsnap vg0  swi-a-s--- 16.00m   lv0    0.07
```



Important - Notez que le snapshot est créé dans le même VG que le LV d'origine.

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

```
SUSE12SP1:~ # lvs --help
lvs: Display information about logical volumes

lvs
[-a|--all]
[--aligned]
[--binary]
[--commandprofile ProfileName]
[-d|--debug]
[--foreign]
[-h|--help]
[--ignorelockingfailure]
[--ignorestalecluster]
[--nameprefixes]
[--noheadings]
[--nosuffix]
[-o|--options [+]Field[,Field]]
[-O|--sort [+]-key1[,[+]-key2[,...]]]
[-P|--partial]
[--readonly]
[--rows]
[--segments]
[-S|--select Selection]
[--separator Separator]
[--trustcache]
[--unbuffered]
[--units hHbBsSkKmMgGtTpPeE]
[--unquoted]
[-v|--verbose]
[--version]
[LogicalVolume[Path] [LogicalVolume[Path]...]]
```


Créez maintenant un répertoire pour monter le snapshot :

```
SUSE12SP1:~ # mkdir /mnt/testsnap
```

Montez le snapshot :

```
SUSE12SP1:~ # mount /dev/vg0/testsnap /mnt/testsnap
```

Comparez le volume d'origine et le snapshot :

```
SUSE12SP1:~ # ls -l /mnt/lvm
total 10294
-rw-r--r-- 1 root root 10485760 Oct 10 17:07 10M
drwxr-xr-x 3 root root    1024 Oct  9 11:57 home
drwx----- 2 root root    12288 Oct 10 16:38 lost+found
SUSE12SP1:~ # ls -l /mnt/testsnap
total 10294
-rw-r--r-- 1 root root 10485760 Oct 10 17:07 10M
drwxr-xr-x 3 root root    1024 Oct  9 11:57 home
drwx----- 2 root root    12288 Oct 10 16:38 lost+found
```

Supprimez maintenant le fichier **10M** de votre volume d'origine :

```
SUSE12SP1:~ # rm /mnt/lvm/10M
```

Constatez le résultat de cette suppression :

```
SUSE12SP1:~ # df -Ph /mnt/lvm
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/vg0-lv0 434M  2.8M  410M   1% /mnt/lvm
SUSE12SP1:~ #
SUSE12SP1:~ # df -Ph /mnt/testsnap
Filesystem      Size  Used Avail Use% Mounted on
```

```
/dev/mapper/vg0-testsnap 434M 13M 400M 4% /mnt/testsnap
SUSE12SP1:~ #
SUSE12SP1:~ # lvs
  LV      VG   Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  lv0     vg0  owi-aos--- 456.00m
  testsnap vg0  swi-aos--- 16.00m      lv0    0.32
```



A faire - Restaurez le fichier 10M à partir du snapshot.

LAB #6 - Suppression des Volumes

La suppression d'un volume logique se fait grâce à la commande **lvremove** :

```
SUSE12SP1:~ # umount /mnt/testsnap/
SUSE12SP1:~ # lvremove /dev/vg0/testsnap
Do you really want to remove active logical volume testsnap? [y/n]: y
  Logical volume "testsnap" successfully removed
SUSE12SP1:~ # umount /mnt/lvm
SUSE12SP1:~ # lvremove /dev/vg0/lv0
Do you really want to remove active logical volume lv0? [y/n]: y
  Logical volume "lv0" successfully removed
```



Important - Notez que cette opération nécessite à ce que le volume logique soit démonté.

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

```
SUSE12SP1:~ # lvremove --help
```

```
lvremove: Remove logical volume(s) from the system
```

```
lvremove
```

```
[-A|--autobackup y|n]  
[--commandprofile ProfileName]  
[-d|--debug]  
[-f|--force]  
[-h|--help]  
[--noudevsync]  
[-S|--select Selection]  
[-t|--test]  
[-v|--verbose]  
[--version]  
LogicalVolume[Path] [LogicalVolume[Path]...]
```

Le Volume Group peut aussi être supprimé :

```
SUSE12SP1:~ # vgrename vg0  
Volume group "vg0" successfully removed
```

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

```
SUSE12SP1:~ # vgrename --help  
vgrename: Remove volume group(s)
```

```
vgrename
```

```
[--commandprofile ProfileName]  
[-d|--debug]  
[-f|--force]  
[-h|--help]  
[--noudevsync]  
[-S|--select Selection]  
[-t|--test]  
[-v|--verbose]
```

```
[--version]  
VolumeGroupName [VolumeGroupName...]
```

Ainsi que le volume physique :

```
SUSE12SP1:~ # pvremove /dev/sdb2 /dev/sdb3 /dev/sdb6  
Labels on physical volume "/dev/sdb2" successfully wiped  
Labels on physical volume "/dev/sdb3" successfully wiped  
Labels on physical volume "/dev/sdb6" successfully wiped
```

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

```
SUSE12SP1:~ # pvremove --help  
pvremove: Remove LVM label(s) from physical volume(s)  
  
pvremove  
  [--commandprofile ProfileName]  
  [-d|--debug]  
  [-f[f]|--force [--force]]  
  [-h|-?|--help]  
  [-t|--test]  
  [-v|--verbose]  
  [--version]  
  [-y|--yes]  
  PhysicalVolume [PhysicalVolume...]
```

LAB #7 - Volumes Logiques en Miroir

Re-créez maintenant votre Volume Group :

```
SUSE12SP1:~ # pvcreate /dev/sdb2 /dev/sdb3 /dev/sdb6  
Physical volume "/dev/sdb2" successfully created
```

```
Physical volume "/dev/sdb3" successfully created
Physical volume "/dev/sdb6" successfully created
SUSE12SP1:~ # vgcreate -s 8M vg0 /dev/sdb2 /dev/sdb3 /dev/sdb6
Volume group "vg0" successfully created
```

Créez maintenant un Logical Volume en miroir grâce à l'option **-m** de la commande **lvcreate**, suivi du nombre de miroirs :

```
SUSE12SP1:~ # lvcreate -m 1 -L 100M -n lv1 vg0
Rounding up size to full physical extent 104.00 MiB
Logical volume "lv1" created.
```

Constatez maintenant la présence du miroir :

```
SUSE12SP1:~ # lvdisplay -m /dev/vg0/lv1
--- Logical volume ---
LV Path                /dev/vg0/lv1
LV Name                 lv1
VG Name                 vg0
LV UUID                 GACvjV-rvto-MBTS-xnsn-7A3e-40JC-vLWh13
LV Write Access         read/write
LV Creation host, time SUSE12SP1.fenestros.loc, 2017-10-11 09:49:30 +0200
LV Status                available
# open                  0
LV Size                 104.00 MiB
Current LE              13
Mirrored volumes        2
Segments                1
Allocation               inherit
Read ahead sectors      auto
- currently set to     1024
Block device            254:4
--- Segments ---
Logical extents 0 to 12:
Type                    raid1
```

```
Monitoring      monitored
Raid Data LV 0
  Logical volume  lv1_rimage_0
  Logical extents 0 to 12
Raid Data LV 1
  Logical volume  lv1_rimage_1
  Logical extents 0 to 12
Raid Metadata LV 0  lv1_rmeta_0
Raid Metadata LV 1  lv1_rmeta_1
```

Le miroir s'étend sur plusieurs volumes physiques :

```
SUSE12SP1:~ # pvdisplay -m /dev/sdb2 /dev/sdb3 /dev/sdb6
--- Physical volume ---
PV Name           /dev/sdb2
VG Name           vg0
PV Size           200.00 MiB / not usable 8.00 MiB
Allocatable       yes
PE Size           8.00 MiB
Total PE          24
Free PE           10
Allocated PE      14
PV UUID           408GSr-a47G-cjB-Vrpd-0fXs-pqZs-l0A7pW
--- Physical Segments ---
Physical extent 0 to 0:
  Logical volume  /dev/vg0/lv1_rmeta_0
  Logical extents 0 to 0
Physical extent 1 to 13:
  Logical volume  /dev/vg0/lv1_rimage_0
  Logical extents 0 to 12
Physical extent 14 to 23:
  FREE
--- Physical volume ---
PV Name           /dev/sdb3
```

```
VG Name          vg0
PV Size          300.00 MiB / not usable 4.00 MiB
Allocatable     yes
PE Size         8.00 MiB
Total PE        37
Free PE         23
Allocated PE    14
PV UUID         YU3YJD-s1nB-wAYz-e329-UcgK-je8c-aXwngf
--- Physical Segments ---
Physical extent 0 to 0:
  Logical volume /dev/vg0/lv1_rmeta_1
  Logical extents 0 to 0
Physical extent 1 to 13:
  Logical volume /dev/vg0/lv1_rimage_1
  Logical extents 0 to 12
Physical extent 14 to 36:
  FREE
--- Physical volume ---
PV Name          /dev/sdb6
VG Name          vg0
PV Size          400.00 MiB / not usable 8.00 MiB
Allocatable     yes
PE Size         8.00 MiB
Total PE        49
Free PE         49
Allocated PE    0
PV UUID         QgKnsV-ZW30-YEaw-LNfY-ZyTt-WXHw-D8StdG
--- Physical Segments ---
Physical extent 0 to 48:
  FREE
```

La suppression du miroir se fait en utilisant la commande **lvconvert** en indiquant quel volume physique doit être vidé de son contenu :

```
SUSE12SP1:~ # lvconvert -m 0 /dev/vg0/lv1 /dev/sdb3
```

De même, il est possible de créer un miroir pour un volume logique existant :

```
SUSE12SP1:~ # lvconvert -m 1 /dev/vg0/lv1
```

Supprimez de nouveau votre miroir :

```
SUSE12SP1:~ # lvconvert -m 0 /dev/vg0/lv1 /dev/sdb3
```

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

```
SUSE12SP1:~ # lvconvert --help
lvconvert: Change logical volume layout

lvconvert [-m|--mirrors Mirrors [{--mirrorlog {disk|core|mirrored}|--corelog}]]
  [--type SegmentType]
  [--repair [--use-policies]]
  [--replace PhysicalVolume]
  [-R|--regionsize MirrorLogRegionSize]
  [--alloc AllocationPolicy]
  [-b|--background]
  [-f|--force]
  [-i|--interval seconds]
  [--stripes Stripes [-I|--stripesize StripeSize]]
  [--commandprofile ProfileName] [-d|--debug] [-h|-?|--help]
  [--noudevsync] [-t|--test] [-v|--verbose] [--version] [-y|--yes]
  LogicalVolume[Path] [PhysicalVolume[Path]...]

lvconvert [--splitmirrors Images --trackchanges]
  [--splitmirrors Images --name SplitLogicalVolumeName]
  [--commandprofile ProfileName] [-d|--debug] [-h|-?|--help]
  [--noudevsync] [-t|--test] [-v|--verbose] [--version] [-y|--yes]
  LogicalVolume[Path] [SplittablePhysicalVolume[Path]...]

lvconvert --splitsnapshot
```



```
[--commandprofile ProfileName] [-d|--debug] [-h|-?|--help]
[--noudevsync] [-t|--test] [-v|--verbose] [--version] [-y|--yes]
SnapshotLogicalVolume[Path]
```

lvconvert --splitcache

```
[--commandprofile ProfileName] [-d|--debug] [-h|-?|--help]
[--noudevsync] [-t|--test] [-v|--verbose] [--version] [-y|--yes]
CacheLogicalVolume[Path]
```

lvconvert --split

```
[--name SplitLogicalVolumeName]
[--commandprofile ProfileName] [-d|--debug] [-h|-?|--help]
[--noudevsync] [-t|--test] [-v|--verbose] [--version] [-y|--yes]
SplittableLogicalVolume[Path]
```

lvconvert --uncache

```
[--commandprofile ProfileName] [-d|--debug] [-h|-?|--help]
[--noudevsync] [-t|--test] [-v|--verbose] [--version] [-y|--yes]
CacheLogicalVolume[Path]
```

lvconvert [--type snapshot|-s|--snapshot]

```
[-c|--chunksize]
[-Z|--zero {y|n}]
[--commandprofile ProfileName] [-d|--debug] [-h|-?|--help]
[--noudevsync] [-t|--test] [-v|--verbose] [--version] [-y|--yes]
OriginalLogicalVolume[Path] SnapshotLogicalVolume[Path]
```

lvconvert --merge

```
[-b|--background]
[-i|--interval seconds]
[--commandprofile ProfileName] [-d|--debug] [-h|-?|--help]
[--noudevsync] [-t|--test] [-v|--verbose] [--version] [-y|--yes]
LogicalVolume[Path]
```

```
lvconvert [--type thin[-pool]]|-T|--thin]
  [--thinpool ThinPoolLogicalVolume[Path]]
  [--chunksize size]
  [--discards {ignore|nopassdown|passdown}]
  [--poolmetadataspare {y|n}]
  [{--poolmetadata ThinMetadataLogicalVolume[Path] |
  --poolmetadatasize size}]
  [-r|--readahead ReadAheadSectors|auto|none]
  [--stripes Stripes [-I|--stripesize StripeSize]]]
  [--originname NewExternalOriginVolumeName]]
  [-Z|--zero {y|n}]
  [--commandprofile ProfileName] [-d|--debug] [-h|-?|--help]
  [--noudevsync] [-t|--test] [-v|--verbose] [--version] [-y|--yes]
  [ExternalOrigin|ThinDataPool]LogicalVolume[Path] [PhysicalVolumePath...]
```

```
lvconvert [--type cache[-pool]]|-H|--cache]
  [--cachepool CacheDataLogicalVolume[Path]]
  [--cachemode CacheMode]
  [--chunksize size]
  [--poolmetadataspare {y|n}]]
  [{--poolmetadata CacheMetadataLogicalVolume[Path] |
  --poolmetadatasize size}]
  [--commandprofile ProfileName] [-d|--debug] [-h|-?|--help]
  [--noudevsync] [-t|--test] [-v|--verbose] [--version] [-y|--yes]
  [Cache|CacheDataPool]LogicalVolume[Path] [PhysicalVolumePath...]
```

LAB #8 - Modifier les Attributs LVM

En cas de présence d'un miroir, la commande **lvs** indique la présence du miroir dans la colonne **Attr** avec la lettre **m** :

```
SUSE12SP1:~ # lvs
  LV   VG   Attr          LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
```

```
lv1  vg0  -wi-a----- 104.00m
```



A faire - Consultez [cette page](#) pour comprendre les attributs.

La commande équivalente pour les Volume Groups est **vgs** :

```
SUSE12SP1:~ # vgs
VG   #PV #LV #SN Attr   VSize   VFree
vg0   3   1   0 wz--n- 880.00m 776.00m
```



A faire - Consultez [cette page](#) pour comprendre les attributs.

La commande équivalente pour les Physical Volumes est **pvs** :

```
SUSE12SP1:~ # pvs
PV          VG   Fmt Attr PSize   PFree
/dev/sdb2  vg0  lvm2 a--  192.00m 88.00m
/dev/sdb3  vg0  lvm2 a--  296.00m 296.00m
/dev/sdb6  vg0  lvm2 a--  392.00m 392.00m
```



A faire - Consultez [cette page](#) pour comprendre les attributs.

Les commandes **lvchange**, **vgchange** et **pvchange** permettent de modifier les attributs des Logical Volumes, Volume Groups et Physical Volumes respectivement.

Par exemple, pour rendre inutilisable un Logical Volume, il convient d'enlever l'attribut **a** :

```
SUSE12SP1:~ # lvchange -a n /dev/vg0/lv1
SUSE12SP1:~ # lvs
  LV   VG   Attr           LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  lv1  vg0  -wi----- 104.00m
```

Pour faire l'inverse il convient de saisir la commande suivante :

```
SUSE12SP1:~ # lvchange -a y /dev/vg0/lv1
SUSE12SP1:~ # lvs
  LV   VG   Attr           LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  lv1  vg0  -wi-a----- 104.00m
```

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

```
SUSE12SP1:~ # lvchange --help
lvchange: Change the attributes of logical volume(s)

lvchange
  [-A|--autobackup y|n]
  [-a|--activate [a|e|l]{y|n}]
  [--activationmode {complete|degraded|partial}    [--addtag Tag]
  [--alloc AllocationPolicy]
  [-C|--contiguous y|n]
  [--cachepolicy policyname] [--cachesettings parameter=value]
  [--commandprofile ProfileName]
  [-d|--debug]
  [--deltag Tag]
  [--detachprofile]
  [--errorwhenfull {y|n}]
  [-f|--force]
  [-h|--help]
  [--discards {ignore|nopassdown|passdown}]
  [--ignorelockingfailure]
  [--ignoremonitoring]
```

```
[--ignoreskippedcluster]
[-k|--setactivationskip {y|n}]
[-K|--ignoreactivationskip]
[--monitor {y|n}]
[--poll {y|n}]
[--noudevsync]
[-M|--persistent y|n] [-j|--major major] [--minor minor]
[--metadataprofile ProfileName]
[-P|--partial]
[-p|--permission r|rw]
[--[raid]minrecoveryrate Rate]
[--[raid]maxrecoveryrate Rate]
[--[raid]syncaction {check|repair}]
[--[raid]writebehind IOCount]
[--[raid]writemostly PhysicalVolume[:{t|n|y}]]
[-r|--readahead ReadAheadSectors|auto|none]
[--refresh]
[--resync]
[-S|--select Selection]
[--sysinit]
[-t|--test]
[-v|--verbose]
[--version]
[-y|--yes]
[-Z|--zero {y|n}]
LogicalVolume[Path] [LogicalVolume[Path]...]
```

LAB #9 - Volumes Logiques en Bandes

Un volume logique en bandes est créé pour augmenter, comme dans le cas du RAID, les performances des entrées et sorties. Pour créer ce volume, la commande **lvcreate** prend deux option supplémentaires :

- **-i** - indique le nombre de volumes de bandes,

- **-l** - indique la taille en Ko de chaque bande.

Saisissez donc la commande suivante :

```
SUSE12SP1:~ # lvcreate -i2 -I64 -n lv2 -L 100M vg0 /dev/sdb3 /dev/sdb6
Rounding up size to full physical extent 104.00 MiB
Rounding size (13 extents) up to stripe boundary size (14 extents).
Logical volume "lv2" created.
```

Constatez la présence de vos bandes sur /dev/sda7 et sur /dev/sda9 :

```
SUSE12SP1:~ # lvsdisplay -m /dev/vg0/lv2
--- Logical volume ---
LV Path                /dev/vg0/lv2
LV Name                lv2
VG Name                vg0
LV UUID                PQ175n-zCmp-78Yi-t2dz-Y0v5-1Waq-K4NV0F
LV Write Access        read/write
LV Creation host, time SUSE12SP1.fenestros.loc, 2017-10-11 09:59:20 +0200
LV Status              available
# open                 0
LV Size                112.00 MiB
Current LE             14
Segments               1
Allocation             inherit
Read ahead sectors    auto
- currently set to    1024
Block device           254:1
--- Segments ---
Logical extents 0 to 13:
  Type                striped
  Stripes              2
  Stripe size         64.00 KiB
  Stripe 0:
```

```
Physical volume  /dev/sdb3
Physical extents 0 to 6
Stripe 1:
Physical volume  /dev/sdb6
Physical extents 0 to 6
```

Utilisez maintenant la commande **lvs** pour visualiser les volumes physiques utilisés par le volume logique :

```
SUSE12SP1:~ # lvs -o +devices
LV  VG  Attr      LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert Devices
lv1  vg0 -wi-a----- 104.00m
lv2  vg0 -wi-a----- 112.00m                               /dev/sdb3(0),/dev/sdb6(0)
```

LAB #10 - Gérer les Métadonnées

Les métadonnées pour chaque Volume Group sont stockés dans un fichier texte au nom du Volume Group dans le répertoire **/etc/lvm/backup** :

```
SUSE12SP1:~ # cat /etc/lvm/backup/vg0
# Generated by LVM2 version 2.02.120(2) (2015-05-15): Wed Oct 11 09:59:20 2017

contents = "Text Format Volume Group"
version = 1

description = "Created *after* executing 'lvcreate -i2 -I64 -n lv2 -L 100M vg0 /dev/sdb3 /dev/sdb6'"

creation_host = "SUSE12SP1.fenestros.loc" # Linux SUSE12SP1.fenestros.loc 3.12.74-60.64.40-default #1 SMP Wed May
10 05:07:16 UTC 2017 (4eb35ec) x86_64
creation_time = 1507708760 # Wed Oct 11 09:59:20 2017

vg0 {
    id = "qCdv6F-cMd5-j2de-u91l-AgpE-3NCQ-J6G33P"
    seqno = 10
    format = "lvm2"          # informational
```

```
status = ["RESIZEABLE", "READ", "WRITE"]
flags = []
extent_size = 16384      # 8 Megabytes
max_lv = 0
max_pv = 0
metadata_copies = 0

physical_volumes {

    pv0 {
        id = "408GSr-a47G-cjbB-Vrpd-OfXs-pqZs-l0A7pW"
        device = "/dev/sdb2"    # Hint only

        status = ["ALLOCATABLE"]
        flags = []
        dev_size = 409600    # 200 Megabytes
        pe_start = 2048
        pe_count = 24    # 192 Megabytes
    }

    pv1 {
        id = "YU3YJD-s1nB-wAYz-e329-UcgK-je8c-aXwngf"
        device = "/dev/sdb3"    # Hint only

        status = ["ALLOCATABLE"]
        flags = []
        dev_size = 614400    # 300 Megabytes
        pe_start = 2048
        pe_count = 37    # 296 Megabytes
    }

    pv2 {
        id = "QgKnsV-ZW30-YEaw-LNfY-ZyTt-WXHw-D8StdG"
        device = "/dev/sdb6"    # Hint only
    }
}
```



```
    status = ["ALLOCATABLE"]
    flags = []
    dev_size = 819200    # 400 Megabytes
    pe_start = 2048
    pe_count = 49    # 392 Megabytes
  }
}

logical_volumes {

  lv1 {
    id = "GACvjV-rvto-MBTS-xnsn-7A3e-40JC-vLWh13"
    status = ["READ", "WRITE", "VISIBLE"]
    flags = []
    creation_host = "SUSE12SP1.fenestros.loc"
    creation_time = 1507708170    # 2017-10-11 09:49:30 +0200
    segment_count = 1

    segment1 {
      start_extent = 0
      extent_count = 13    # 104 Megabytes

      type = "striped"
      stripe_count = 1    # linear

      stripes = [
        "pv0", 1
      ]
    }
  }

  lv2 {
    id = "PQ175n-zCmp-78Yi-t2dz-Y0v5-1Waq-K4NV0F"
    status = ["READ", "WRITE", "VISIBLE"]
```

```
flags = []
creation_host = "SUSE12SP1.fenestros.loc"
creation_time = 1507708760    # 2017-10-11 09:59:20 +0200
segment_count = 1

segment1 {
    start_extent = 0
    extent_count = 14    # 112 Megabytes

    type = "striped"
    stripe_count = 2
    stripe_size = 128    # 64 Kilobytes

    stripes = [
        "pv1", 0,
        "pv2", 0
    ]
}
}
}
```

Des archives sont créées lors de chaque modification d'un groupe de volumes et elles sont placés dans le répertoire **/etc/lvm/archives** :

```
SUSE12SP1:~ # ls /etc/lvm/archive/
vg0_00000-1490864187.vg  vg0_00005-1861990860.vg  vg0_00010-1593394958.vg
vg0_00001-1258721014.vg  vg0_00006-475023216.vg  vg0_00011-1525730672.vg
vg0_00002-1552831049.vg  vg0_00007-1421103332.vg  vg0_00012-1449042750.vg
vg0_00003-1050448537.vg  vg0_00008-577028682.vg
vg0_00004-1537549129.vg  vg0_00009-1035978808.vg
```

La commande **vgcfgbackup** est utilisée pour sauvegarder les métadonnées manuellement dans le fichier **/etc/lvm/backup/nom_du_volume_group** :

La commande **vgcfgrestore** permet de restaurer une sauvegarde. Sans l'option **-f** pour spécifier la sauvegarde à restaurer, cette commande renvoie la liste des sauvegardes disponibles :

```
SUSE12SP1:~ # vgcfgbackup vg0
Volume group "vg0" successfully backed up.
```

Il est aussi possible de modifier l'emplacement de la sauvegarde avec l'option **-f** de la commande :

```
SUSE12SP1:~ # vgcfgbackup -f /tmp/vg0_backup vg0
Volume group "vg0" successfully backed up.
```

```
SUSE12SP1:~ # vgcfgrestore --list vg0
File:      /etc/lvm/archive/vg0_00000-1490864187.vg
Couldn't find device with uuid SdDEnc-2fiL-IY82-K0zp-MScx-w1Nj-zefWFA.
Couldn't find device with uuid lnUsag-cM0o-76LP-PffQ-wJQK-0NKa-vjtFzy.
Couldn't find device with uuid f006lr-02qe-ccmx-4JV6-98zm-CGBr-EWQUho.
VG name:   vg0
Description: Created *before* executing 'vgcreate -s 8M vg0 /dev/sdb2 /dev/sdb3 /dev/sdb6'
Backup Time: Tue Oct 10 16:31:14 2017

File:      /etc/lvm/archive/vg0_00001-1258721014.vg
VG name:   vg0
Description: Created *before* executing 'lvcreate -L 350 -n lv0 vg0'
Backup Time: Tue Oct 10 16:34:48 2017

File:      /etc/lvm/archive/vg0_00002-1552831049.vg
VG name:   vg0
Description: Created *before* executing 'lvextend -L +100M /dev/vg0/lv0'
Backup Time: Tue Oct 10 16:44:21 2017

File:      /etc/lvm/archive/vg0_00003-1050448537.vg
VG name:   vg0
Description: Created *before* executing 'lvcreate -s -L 15M -n testsnap /dev/vg0/lv0'
Backup Time: Tue Oct 10 17:09:48 2017
```

```
File:      /etc/lvm/archive/vg0_00004-1537549129.vg
VG name:   vg0
Description: Created *before* executing 'lvremove /dev/vg0/testsnap'
Backup Time: Wed Oct 11 09:45:08 2017
```

```
File:      /etc/lvm/archive/vg0_00005-1861990860.vg
VG name:   vg0
Description: Created *before* executing 'lvremove /dev/vg0/lv0'
Backup Time: Wed Oct 11 09:45:41 2017
```

```
File:      /etc/lvm/archive/vg0_00006-475023216.vg
VG name:   vg0
Description: Created *before* executing 'vgremove vg0'
Backup Time: Wed Oct 11 09:46:42 2017
```

```
File:      /etc/lvm/archive/vg0_00007-1421103332.vg
Couldn't find device with uuid 408GSr-a47G-cjbB-Vrpd-0fXs-pqZs-l0A7pW.
Couldn't find device with uuid YU3YJD-s1nB-wAYz-e329-UcgK-je8c-aXwngf.
Couldn't find device with uuid QgKnsV-ZW30-YEaw-LNfY-ZyTt-WXHw-D8StdG.
VG name:   vg0
Description: Created *before* executing 'vgcreate -s 8M vg0 /dev/sdb2 /dev/sdb3 /dev/sdb6'
Backup Time: Wed Oct 11 09:49:08 2017
```

```
File:      /etc/lvm/archive/vg0_00008-577028682.vg
VG name:   vg0
Description: Created *before* executing 'lvcreate -m 1 -L 100M -n lv1 vg0'
Backup Time: Wed Oct 11 09:49:30 2017
```

```
File:      /etc/lvm/archive/vg0_00009-1035978808.vg
VG name:   vg0
Description: Created *before* executing 'lvconvert -m 0 /dev/vg0/lv1 /dev/sdb3'
Backup Time: Wed Oct 11 09:51:45 2017
```

```
File:      /etc/lvm/archive/vg0_00010-1593394958.vg
```

```
VG name:      vg0
Description:  Created *before* executing 'lvconvert -m 1 /dev/vg0/lv1'
Backup Time:  Wed Oct 11 09:52:14 2017

File:        /etc/lvm/archive/vg0_00011-1525730672.vg
VG name:      vg0
Description:  Created *before* executing 'lvconvert -m 0 /dev/vg0/lv1 /dev/sdb3'
Backup Time:  Wed Oct 11 09:52:18 2017

File:        /etc/lvm/archive/vg0_00012-1449042750.vg
VG name:      vg0
Description:  Created *before* executing 'lvcreate -i2 -I64 -n lv2 -L 100M vg0 /dev/sdb3 /dev/sdb6'
Backup Time:  Wed Oct 11 09:59:20 2017

File:        /etc/lvm/backup/vg0
VG name:      vg0
Description:  Created *after* executing 'vgcfgbackup vg0'
Backup Time:  Wed Oct 11 10:01:43 2017
```

Systemes 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
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](#)

Gestion d'Ext3

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

```
SUSE12SP1:~ # fdisk -l
```

```
Disk /dev/sdb: 8 GiB, 8589934592 bytes, 16777216 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
Disklabel type: dos
Disk identifier: 0x31bcc5e2

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	1026047	1024000	500M	fd	Linux raid autodetect
/dev/sdb2		1026048	1435647	409600	200M	8e	Linux LVM
/dev/sdb3		1435648	2050047	614400	300M	8e	Linux LVM
/dev/sdb4		2050048	16777215	14727168	7G	5	Extended
/dev/sdb5		2052096	3076095	1024000	500M	fd	Linux raid autodetect
/dev/sdb6		3078144	3897343	819200	400M	8e	Linux LVM
/dev/sdb7		3899392	4923391	1024000	500M	fd	Linux raid autodetect
/dev/sdb8		4925440	5949439	1024000	500M	fd	Linux raid autodetect
/dev/sdb9		5951488	6361087	409600	200M	83	Linux

Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 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
Disklabel type: dos
Disk identifier: 0x000cd018

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sda1		2048	4208639	4206592	2G	82	Linux swap / Solaris
/dev/sda2	*	4208640	41943039	37734400	18G	83	Linux

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 notre cas, il s'agit de **/dev/sdb9**.

Créez un filesystem Ext3 sur /dev/sdb9 en utilisant la commande **mke2fs -j** :

```
SUSE12SP1:~ # mke2fs -j /dev/sdb9
mke2fs 1.42.11 (09-Jul-2014)
Creating filesystem with 204800 1k blocks and 51200 inodes
Filesystem UUID: edad27da-4d67-40fb-af9b-d1bace3d1733
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 :

```
SUSE12SP1:~ # 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]
    [-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] [-jnvDFKSV] 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 #11 - Convertir un Système de Fichiers Ext3 en Ext2

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

```
SUSE12SP1:~ # dumpe2fs -h /dev/sdb9
dumpe2fs 1.42.11 (09-Jul-2014)
Filesystem volume name:   <none>
Last mounted on:         <not available>
Filesystem UUID:         edad27da-4d67-40fb-af9b-d1bace3d1733
Filesystem magic number: 0xEF53
Filesystem revision #:   1 (dynamic)
Filesystem features:     has_journal ext_attr resize_inode dir_index filetype sparse_super
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
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: Wed Oct 11 12:52:09 2017
Last mount time: n/a
Last write time: Wed Oct 11 12:52:09 2017
Mount count: 0
Maximum mount count: -1
Last checked: Wed Oct 11 12:52:09 2017
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: 020741aa-ed55-4a18-b403-9605d7c95051
Journal backup: inode blocks
Journal features: (none)
Journal size: 4113k
Journal length: 4096
Journal sequence: 0x00000001
Journal start: 0
```



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

Les options de cette commande sont :

```
SUSE12SP1:~ # dumpe2fs --help
dumpe2fs 1.42.11 (09-Jul-2014)
dumpe2fs: invalid option -- '-'
```

```
Usage: dumpe2fs [-bfhixV] [-o superblock=<num>] [-o blocksize=<num>] device
```

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

```
SUSE12SP1:~ # tune2fs -0 ^has_journal /dev/sdb9
tune2fs 1.42.11 (09-Jul-2014)
```

Les options de cette commande sont :

```
SUSE12SP1:~ # tune2fs --help
tune2fs 1.42.11 (09-Jul-2014)
tune2fs: invalid option -- '-'
Usage: tune2fs [-c max_mounts_count] [-e errors_behavior] [-g group]
      [-i interval[d|m|w]] [-j] [-J journal_options] [-l]
      [-m reserved_blocks_percent] [-o [^]mount_options[,...]] [-p mmp_update_interval]
      [-r reserved_blocks_count] [-u user] [-C mount_count] [-L volume_label]
      [-M last_mounted_dir] [-O [^]feature[,...]]
      [-E extended-option[,...]] [-T last_check_time] [-U UUID]
      [ -I new_inode_size ] device
```

Constatez le résultat de cette commande :

```
SUSE12SP1:~ # dumpe2fs -h /dev/sdb9
dumpe2fs 1.42.11 (09-Jul-2014)
Filesystem volume name:   <none>
Last mounted on:         <not available>
Filesystem UUID:         edad27da-4d67-40fb-af9b-d1bace3d1733
Filesystem magic number: 0xEF53
Filesystem revision #:   1 (dynamic)
Filesystem features:     ext_attr resize_inode dir_index filetype sparse_super
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
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:     Wed Oct 11 12:52:09 2017
Last mount time:        n/a
Last write time:        Wed Oct 11 12:55:10 2017
Mount count:           0
Maximum mount count:    -1
Last checked:           Wed Oct 11 12:52:09 2017
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:    020741aa-ed55-4a18-b403-9605d7c95051
Journal backup:         inode blocks
```



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

Supprimez maintenant l'inode du journal :

```
SUSE12SP1:~ # fsck /dev/sdb9
fsck from util-linux 2.25
e2fsck 1.42.11 (09-Jul-2014)
/dev/sdb9: clean, 11/51200 files, 8013/204800 blocks
```

Créez un point de montage pour /dev/sdb9 :

```
SUSE12SP1:~ # mkdir /mnt/sdb9
```

Essayez de monter /dev/sdb9 en tant que système de fichiers Ext3. Vous obtiendrez un résultat similaire à celui-ci :

```
SUSE12SP1:~ # mount -t ext3 /dev/sdb9 /mnt/sdb9
mount: wrong fs type, bad option, bad superblock on /dev/sdb9,
missing codepage or helper program, or other error

In some cases useful info is found in syslog - try
dmesg | tail or so.
```



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 :

```
SUSE12SP1:~ # mount -t ext2 /dev/sdb9 /mnt/sdb9
SUSE12SP1:~ #
```

LAB #12 - Convertir un Système de Fichiers Ext2 en Ext3

Pour remplacer le journal sur /dev/sdb9, il convient d'utiliser la commande **tune2fs** :

```
SUSE12SP1:~ # umount /mnt/sdb9
SUSE12SP1:~ #
SUSE12SP1:~ # tune2fs -j /dev/sdb9
tune2fs 1.42.11 (09-Jul-2014)
Creating journal inode: done
```



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

LAB #13 - 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/sdb8 :

```
SUSE12SP1:~ # mke2fs -O journal_dev /dev/sdb8
mke2fs 1.42.11 (09-Jul-2014)
Creating filesystem with 512000 1k blocks and 0 inodes
Filesystem UUID: 9056ef8b-ceed-4e54-91cb-0d292742b803
Superblock backups stored on blocks:

Zeroing journal device: SUSE12SP1:~ #
```



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

Créez maintenant un système de fichiers Ext3 sur /dev/sdb9 en plaçant le journal sur /dev/sdb8 :

```
SUSE12SP1:~ # mke2fs -j -J device=/dev/sdb8 /dev/sdb9
mke2fs 1.42.11 (09-Jul-2014)
Using journal device's blocksize: 1024
/dev/sdb9 contains a ext3 file system
    last mounted on Wed Oct 11 13:31:57 2017
Proceed anyway? (y,n) y
Creating filesystem with 204800 1k blocks and 51200 inodes
Filesystem UUID: 2eb54210-df3c-4257-9764-2f6e8c4c1b33
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Adding journal to device /dev/sdb8: done
Writing superblocks and filesystem accounting information: done
```



Important : Notez que le journal a été placé sur /dev/sdb8 grâce à l'utilisation de l'option **-J**.

LAB #14 - Modifier la Fréquence de Vérification du Système de Fichiers Ext3

Pour modifier la fréquence de vérification du système de fichiers sur /dev/sdb9, il convient d'utiliser soit l'option **-c**, soit l'option **-i** :

```
SUSE12SP1:~ # tune2fs -i 100d /dev/sdb9
tune2fs 1.42.11 (09-Jul-2014)
Setting interval between checks to 8640000 seconds
```

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

```
SUSE12SP1:~ # dumpe2fs /dev/sdb9 | grep UUID
```

```
dumpe2fs 1.42.11 (09-Jul-2014)
Filesystem UUID:      2eb54210-df3c-4257-9764-2f6e8c4c1b33
Journal UUID:        9056ef8b-ceed-4e54-91cb-0d292742b803
```

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 #15 - Créer un Système de Fichiers Ext4

Créez un système de fichiers Ext4 sur **/dev/sdb8** :

```
SUSE12SP1:~ # mkfs.ext4 /dev/sdb8
mke2fs 1.42.11 (09-Jul-2014)
/dev/sdb8 contains a jbd file system
```



```
Proceed anyway? (y,n) y
Creating filesystem with 512000 1k blocks and 128016 inodes
Filesystem UUID: ab2093f4-4e46-44cd-b73c-37a6e3d7849b
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 :

```
SUSE12SP1:~ # 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]
    [-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] [-jnqvDFKSV] device [blocks-count]
```

Consultez maintenant les caractéristiques du système de fichier :

```
SUSE12SP1:~ # dumpe2fs /dev/sdb8 | more
dumpe2fs 1.42.11 (09-Jul-2014)
Filesystem volume name:   <none>
Last mounted on:         <not available>
Filesystem UUID:         ab2093f4-4e46-44cd-b73c-37a6e3d7849b
Filesystem magic number: 0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index filetype extent flex_bg sparse_super
huge_file uninit_bg dir_nlink extra_isize
```

```
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
Free blocks:          485334
Free inodes:          128005
First block:          1
Block size:           1024
Fragment size:        1024
Reserved GDT blocks:  256
Blocks per group:     8192
Fragments per group: 8192
Inodes per group:     2032
Inode blocks per group: 254
Flex block group size: 16
Filesystem created:   Wed Oct 11 13:42:38 2017
Last mount time:      n/a
Last write time:      Wed Oct 11 13:42:38 2017
Mount count:          0
Maximum mount count:  -1
Last checked:         Wed Oct 11 13:42:38 2017
Check interval:       0 (<none>)
--More--
```

LAB #16 - Ajouter une Etiquette au Système de Fichiers Ext4

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

```
r
```

```
SUSE12SP1:~ # e2label /dev/sdb8 my_ext4
SUSE12SP1:~ # dumpe2fs /dev/sdb8 | grep volume
dumpe2fs 1.42.11 (09-Jul-2014)
Filesystem volume name:  my_ext4
```



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

Créez un point de montage dans **/mnt** et essayez de monter **/dev/sdb8** en tant qu'Ext3 :

```
SUSE12SP1:~ # mkdir /mnt/sdb8
SUSE12SP1:~ #
SUSE12SP1:~ # mount -t ext3 /dev/sdb8 /mnt/sdb8
mount: wrong fs type, bad option, bad superblock on /dev/sdb8,
missing codepage or helper program, or other error

In some cases useful info is found in syslog - try
dmesg | tail or so.
```



Important - Notez l'erreur qui est signalée.

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

```
SUSE12SP1:~ # mount /dev/sdb8 /mnt/sdb8
SUSE12SP1:~ #
SUSE12SP1:~ # mount | grep sdb8
/dev/sdb8 on /mnt/sdb8 type ext4 (rw,relatime,data=ordered)
```





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

LAB #17 - Convertir un Système de Fichiers Ext3 en Ext4

Créez un système de fichiers ext3 sur /dev/sdb9 :

```
SUSE12SP1:~ # mkfs.ext3 /dev/sdb9
mke2fs 1.42.11 (09-Jul-2014)
/dev/sdb9 contains a ext3 file system
   created on Wed Oct 11 13:40:14 2017
Proceed anyway? (y,n) y
Creating filesystem with 204800 1k blocks and 51200 inodes
Filesystem UUID: 299b24aa-d708-4934-b7c6-5741c49374c4
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/sdb9** sur /mnt/sdb9 :

```
SUSE12SP1:~ # mount /dev/sdb9 /mnt/sdb9
SUSE12SP1:~ # ls -l /mnt/sdb9
total 12
drwx----- 2 root root 12288 Oct 11 13:48 lost+found
```

Créez le fichier **/mnt/sdb9/check_file** :

```
SUSE12SP1:~ # touch /mnt/sdb9/check_file
```

Injectez la chaîne **check file** dans le fichier /mnt/sdb9/check_file puis démontez /dev/sdb9 :

```
SUSE12SP1:~ # echo "check file" > /mnt/sdb9/check_file
SUSE12SP1:~ #
SUSE12SP1:~ # umount /dev/sdb9
```

Exécutez e2fsck sur /dev/sdb9 :

```
SUSE12SP1:~ # e2fsck /dev/sdb9
e2fsck 1.42.11 (09-Jul-2014)
/dev/sdb9: clean, 12/51200 files, 12127/204800 blocks
```

Convertissez /dev/sdb9 en Ext4 :

```
SUSE12SP1:~ # tune2fs -0 extents,uninit_bg,dir_index /dev/sdb9
tune2fs 1.42.11 (09-Jul-2014)
```

Optimisez le système de fichiers :

```
SUSE12SP1:~ # e2fsck -fDC0 /dev/sdb9
e2fsck 1.42.11 (09-Jul-2014)
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/sdb9: ***** FILE SYSTEM WAS MODIFIED *****
/dev/sdb9: 12/51200 files (0.0% non-contiguous), 12127/204800 blocks
```

Essayez de monter **/dev/sdb9** en tant qu'Ext3 :

```
SUSE12SP1:~ # mount -t ext3 /dev/sdb9 /mnt/sdb9
```

```
mount: wrong fs type, bad option, bad superblock on /dev/sdb9,  
missing codepage or helper program, or other error
```

```
In some cases useful info is found in syslog - try  
dmesg | tail or so.
```

Montez /dev/sdb9 sans spécifier le type de fichiers Ext3 et vérifiez le contenu du fichier **check_file** :

```
SUSE12SP1:~ # mount /dev/sdb9 /mnt/sdb9  
SUSE12SP1:~ #  
SUSE12SP1:~ # ls -l /mnt/sdb9  
total 13  
-rw-r--r-- 1 root root    11 Oct 11 13:51 check_file  
drwx----- 2 root root 12288 Oct 11 13:48 lost+found  
SUSE12SP1:~ #  
SUSE12SP1:~ # cat /mnt/sdb9/check_file  
check file
```

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

```
SUSE12SP1:~ # dumpe2fs /dev/sdb8 | grep UUID  
dumpe2fs 1.42.11 (09-Jul-2014)  
Filesystem UUID:          ab2093f4-4e46-44cd-b73c-37a6e3d7849b  
SUSE12SP1:~ # dumpe2fs /dev/sdb9 | grep UUID  
dumpe2fs 1.42.11 (09-Jul-2014)  
Filesystem UUID:          299b24aa-d708-4934-b7c6-5741c49374c4
```

XFS

XFS est un système de fichiers 64-bit journalisé de haute performance créée 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.

Debian propose aussi une version 32 bits du système de fichiers XFS.

Pour plus d'informations concernant XFS, consultez [cette page](#).

LAB #18 - Créer un Système de Fichiers XFS

Démontez **/dev/sdb9** :

```
SUSE12SP1:~ # umount /dev/sdb9
```

Créez un système de fichiers XFS sur la partition **/dev/sdb9** :

```
SUSE12SP1:~ # mkfs.xfs /dev/sdb9
mkfs.xfs: /dev/sdb9 appears to contain an existing filesystem (ext4).
mkfs.xfs: Use the -f option to force overwrite.
SUSE12SP1:~ # mkfs.xfs -f /dev/sdb9
meta-data=/dev/sdb9          isize=256    agcount=4, agsize=12800 blks
           =                  sectsz=512   attr=2, projid32bit=1
           =                  crc=0        finobt=0
data      =                  bsize=4096  blocks=51200, imaxpct=25
           =                  sunit=0      swidth=0 blks
naming    =version 2        bsize=4096  ascii-ci=0 ftype=0
log       =internal log    bsize=4096  blocks=853, version=2
           =                  sectsz=512   sunit=0 blks, lazy-count=1
realtime  =none            extsz=4096  blocks=0, rtextents=0
```



Important - Notez l'utilisation de l'option **-f** afin d'écraser le système de fichiers Ext4 existant.

Les options de cette commande sont :

```
SUSE12SP1:~ # mkfs.xfs --help
mkfs.xfs: invalid option -- '-'
```

```
unknown option --
Usage: mkfs.xfs
/* blocksize */      [-b log=n|size=num]
/* metadata */      [-m crc=0|1,finobt=0|1]
/* data subvol */    [-d agcount=n,agsize=n,file,name=xxx,size=num,
                    (sunit=value,swidth=value|su=num,sw=num|noalign),
                    sectlog=n|sectsize=num]
/* force overwrite */ [-f]
/* inode size */     [-i log=n|perblock=n|size=num,maxpct=n,attr=0|1|2,
                    projid32bit=0|1]
/* no discard */     [-K]
/* log subvol */     [-l agnum=n,internal,size=num,logdev=xxx,version=n
                    sunit=value|su=num,sectlog=n|sectsize=num,
                    lazy-count=0|1]
/* label */         [-L label (maximum 12 characters)]
/* naming */         [-n log=n|size=num,version=2|ci,ftype=0|1]
/* no-op info only */ [-N]
/* prototype file */ [-p fname]
/* quiet */         [-q]
/* realtime subvol */ [-r extsize=num,size=num,rtdev=xxx]
/* sectorsize */     [-s log=n|size=num]
/* version */       [-V]
                    devicename
<devicename> is required unless -d name=xxx is given.
<num> is xxx (bytes), xxxs (sectors), xxxb (fs blocks), xxxk (xxx KiB),
      xxxm (xxx MiB), xxxg (xxx GiB), xxxt (xxx TiB) or xxxp (xxx PiB).
<value> is xxx (512 byte blocks).
```

Consultez maintenant les caractéristiques du système de fichier :

```
SUSE12SP1:~ # xfs_info /dev/sdb9
xfs_info: /dev/sdb9 is not a mounted XFS filesystem
SUSE12SP1:~ # mount /dev/sdb9 /mnt/sdb9
SUSE12SP1:~ # xfs_info /dev/sdb9
```



```
meta-data=/dev/sdb9      isize=256   agcount=4, agsize=12800 blks
      =                sectsz=512   attr=2, projid32bit=1
      =                crc=0       finobt=0
data      =                bsize=4096   blocks=51200, imaxpct=25
      =                sunit=0      swidth=0 blks
naming    =version 2     bsize=4096   ascii-ci=0 ftype=0
log       =internal      bsize=4096   blocks=853, version=2
      =                sectsz=512   sunit=0 blks, lazy-count=1
realtime  =none          extsz=4096   blocks=0, rtextents=0
```



Important - Notez que la partition XFS doit être montée pour pouvoir utiliser la commande **xfs_info**.

Les options de cette commande sont :

```
SUSE12SP1:~ # xfs_info --help
/usr/sbin/xfs_info: illegal option -- -
Usage: xfs_info [-V] [-t mtab] mountpoint
```

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

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

```
SUSE12SP1:~ # xfs_admin -L my_xfs /dev/sdb9
xfs_admin: /dev/sdb9 contains a mounted filesystem

fatal error -- couldn't initialize XFS library

SUSE12SP1:~ # umount /dev/sdb9

SUSE12SP1:~ # xfs_admin -L my_xfs /dev/sdb9
```

```
writing all SBs  
new label = "my_xfs"
```



Important - Notez que la partition XFS doit être démonté pour pouvoir utiliser la commande **xfs_admin**.

Pour voir l'étiquette, utilisez la commande suivante :

```
SUSE12SP1:~ # xfs_admin -l /dev/sdb9  
label = "my_xfs"
```



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

Les options de cette commande sont :

```
SUSE12SP1:~ # xfs_admin --help  
/usr/sbin/xfs_admin: illegal option -- -  
Usage: xfs_admin [-efjlpUV] [-c 0|1] [-L label] [-U uuid] device
```

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

```
SUSE12SP1:~ # xfs_admin -u /dev/sdb9  
UUID = 88690d3d-c6de-4b81-88ec-cff03a1de984
```



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](#).

LAB #20 - Créer un Système de Fichiers ReiserFS

Créez un système de fichiers ReiserFS sur la partition **/dev/sda9** :

```
SUSE12SP1:~ # mkfs.reiserfs /dev/sdb7
mkfs.reiserfs 3.6.24

Guessing about desired format.. Kernel 3.12.74-60.64.40-default 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: 0d6ce4b9-9b35-4fa5-8b77-43a058b0704b
ATTENTION: YOU SHOULD REBOOT AFTER FDISK!
    ALL DATA WILL BE LOST ON '/dev/sdb7'!
Continue (y/n):y
Initializing journal - 0%....20%....40%....60%....80%....100%
Syncing..ok
```

```
ReiserFS is successfully created on /dev/sdb7.
```

Les options de cette commande sont :

```
SUSE12SP1:~ # mkfs.reiserfs --help
mkfs.reiserfs: unrecognized option '--help'
Usage: mkfs.reiserfs [options] device [block-count]

Options:

-b | --block-size N           size of file-system block, in bytes
-j | --journal-device FILE    path to separate device to hold journal
-s | --journal-size N        size of the journal in blocks
-o | --journal-offset N      offset of the journal from the start of
                             the separate device, in blocks
-t | --transaction-max-size N maximal size of transaction, in blocks
-B | --badblocks file         store all bad blocks given in file on the fs
-h | --hash rupasov|tea|r5    hash function to use by default
-u | --uuid UUID              store UUID in the superblock
-l | --label LABEL            store LABEL in the superblock
--format 3.5|3.6              old 3.5 format or newer 3.6
-f | --force                  specified once, make mkreiserfs the whole
                             disk, not block device or mounted partition;
                             specified twice, do not ask for confirmation
-q | --quiet                  quiet work without messages, progress and
                             questions. Useful if run in a script. For use
                             by end users only.
-d | --debug                  print debugging information during mkreiser
-V                             print version and exit
```

Consultez maintenant les caractéristiques du système de fichier :

```
SUSE12SP1:~ # debugreiserfs /dev/sdb7
debugreiserfs 3.6.24
```

```
Filesystem state: consistent

Reiserfs super block in block 16 on 0x817 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 [0x40f9e556]
    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: 0d6ce4b9-9b35-4fa5-8b77-43a058b0704b
LABEL:
Set flags in SB:
    ATTRIBUTES CLEAN
Mount count: 1
Maximum mount count: 30
Last fsck run: Wed Oct 11 15:48:41 2017
Check interval in days: 180
```

Les options de cette commande sont :

```
SUSE12SP1:~ # debugreiserfs --help
debugreiserfs: invalid option -- '-'
debugreiserfs: invalid option -- 'h'
debugreiserfs: invalid option -- 'e'
Usage: debugreiserfs [options] device
```

Options:

```
-d          print blocks details of the internal tree
-D          print blocks details of all used blocks
-B file     extract list of badblocks
-m          print bitmap blocks
-o          print objectid map

-J          print journal header
-j filename
            print journal located on the device 'filename'
            stores the journal in the specified file 'filename.'
-p          send filesystem metadata to stdout
-u          read stdin and unpack the metadata
-S          handle all blocks, not only used
-l block   block to print
-q          no speed info
-V          print version and exit
```

LAB #21 - Ajouter une Etiquette au Système de Fichiers ReiserFS

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

```
SUSE12SP1:~ # reiserfstune -l my_reiserfs /dev/sdb7
reiserfstune: Journal device has not been specified. Assuming journal is on the main device (/dev/sdb7).
```

Current parameters:

Filesystem state: consistent

Reiserfs super block in block 16 on 0x817 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 [0x40f9e556]

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: 0d6ce4b9-9b35-4fa5-8b77-43a058b0704b

LABEL: my_reiserfs

Set flags in SB:

ATTRIBUTES CLEAN

Mount count: 1

Maximum mount count: 30

Last fsck run: Wed Oct 11 15:48:41 2017

Check interval in days: 180



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

Les options de cette commande sont :

```
SUSE12SP1:~ # reiserfstune --help
reiserfstune: Usage: reiserfstune [options] device [block-count]

Options:

-j | --journal-device file    current journal device
--journal-new-device file    new journal device
-o | --journal-new-offset N   new journal offset in blocks
-s | --journal-new-size N    new journal size in blocks
-t | --trans-max-size N      new journal max transaction size in blocks
--no-journal-available      current journal is not available
--make-journal-standard     new journal to be standard
-b | --add-badblocks file    add to bad block list
-B | --badblocks file       set the bad block list
-u | --uuid UUID|random     set new UUID
-l | --label LABEL         set new label
-f | --force                force tuning, less confirmations
-c | --check-interval       set interval in days for fsck -a to check,
                           "disable" to disable check,
                           or "default" to restore default
-C | --time-last-checked    set the time the filesystem was last checked
                           (now or YYYYMMDD[HH[MM[SS]]])
-m | --max-mnt-count        set maximum number of mounts before fsck -a
                           checks, "disable" to disable check,
                           or "default" to restore default
-M | --mnt-count            set the number of times the filesystem
                           has been mounted
-h | --help                print help and exit
-V                          print version and exit
```

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


```
SUSE12SP1:~ # debugreiserfs /dev/sdb7 | grep UUID
debugreiserfs 3.6.24
```

```
UUID: 0d6ce4b9-9b35-4fa5-8b77-43a058b0704b
```

JFS

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

Depuis la SLES 11-SP4, SUSE ne supporte plus JFS et a retiré tous les utilitaires de sa distribution.

Pour plus d'informations concernant JFS, consultez [cette page](#).

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 compression du système de fichiers.

LAB #22 - Créer un Système de Fichiers Btrfs

Créez un système de fichiers Btrfs sur **/dev/mapper/vg0-lv1** :

```
SUSE12SP1:~ # mkfs.btrfs /dev/mapper/vg0-lv1
SMALL VOLUME: forcing mixed metadata/data groups
btrfs-progs v4.1.2+20151002
See http://btrfs.wiki.kernel.org for more information.
```

```
Label:                (null)
UUID:                 43e97638-f309-4769-a47a-b5d557f361f8
Node size:            4096
Sector size:          4096
Filesystem size:      104.00MiB
Block group profiles:
  Data+Metadata:      single          8.00MiB
  System:              single          4.00MiB
SSD detected:         no
Incompat features:    mixed-bg, extref, skinny-metadata
Number of devices:    1
Devices:
  ID      SIZE  PATH
  1      104.00MiB /dev/mapper/vg0-lv1
```

Les options de cette commande sont :

```
SUSE12SP1:~ # mkfs.btrfs --help
usage: mkfs.btrfs [options] dev [ dev ... ]
options:
  -A|--alloc-start START  the offset to start the FS
  -b|--byte-count SIZE    total number of bytes in the FS
  -d|--data PROFILE       data profile, raid0, raid1, raid5, raid6, raid10, dup or single
  -f|--force               force overwrite of existing filesystem
  -l|--leafsize SIZE      deprecated, alias for nodesize
  -L|--label LABEL        set a label
  -m|--metadata PROFILE   metadata profile, values like data profile
  -M|--mixed               mix metadata and data together
  -n|--nodesize SIZE      size of btree nodes
```

```
-s|--sectorsize SIZE    min block allocation (may not mountable by current kernel)
-r|--rootdir DIR        the source directory
-K|--nodiscard          do not perform whole device TRIM
-O|--features LIST      comma separated list of filesystem features, use '-O list-all' to list features
-U|--uuid UUID          specify the filesystem UUID
-q|--quiet              no messages except errors
-V|--version            print the mkfs.btrfs version and exit
btrfs-progs v4.1.2+20151002
```

Montez la partition btrfs sur /mnt :

```
SUSE12SP1:~ # mount -t btrfs /dev/vg0/lv1 /mnt/
SUSE12SP1:~ # mount | grep btrfs
/dev/sda2 on / type btrfs (rw,relatime,space_cache,subvolid=259,subvol=@/.snapshots/1/snapshot)
/dev/sda2 on /.snapshots type btrfs (rw,relatime,space_cache,subvolid=258,subvol=@/.snapshots)
/dev/sda2 on /var/spool type btrfs (rw,relatime,space_cache,subvolid=276,subvol=@/var/spool)
/dev/sda2 on /var/opt type btrfs (rw,relatime,space_cache,subvolid=275,subvol=@/var/opt)
/dev/sda2 on /var/lib/named type btrfs (rw,relatime,space_cache,subvolid=272,subvol=@/var/lib/named)
/dev/sda2 on /var/lib/pgsql type btrfs (rw,relatime,space_cache,subvolid=273,subvol=@/var/lib/pgsql)
/dev/sda2 on /var/tmp type btrfs (rw,relatime,space_cache,subvolid=277,subvol=@/var/tmp)
/dev/sda2 on /var/lib/mariadb type btrfs (rw,relatime,space_cache,subvolid=270,subvol=@/var/lib/mariadb)
/dev/sda2 on /var/lib/mysql type btrfs (rw,relatime,space_cache,subvolid=271,subvol=@/var/lib/mysql)
/dev/sda2 on /var/lib/mailman type btrfs (rw,relatime,space_cache,subvolid=269,subvol=@/var/lib/mailman)
/dev/sda2 on /usr/local type btrfs (rw,relatime,space_cache,subvolid=266,subvol=@/usr/local)
/dev/sda2 on /var/crash type btrfs (rw,relatime,space_cache,subvolid=267,subvol=@/var/crash)
/dev/sda2 on /var/lib/libvirt/images type btrfs
(rw,relatime,space_cache,subvolid=268,subvol=@/var/lib/libvirt/images)
/dev/sda2 on /tmp type btrfs (rw,relatime,space_cache,subvolid=265,subvol=@/tmp)
/dev/sda2 on /opt type btrfs (rw,relatime,space_cache,subvolid=263,subvol=@/opt)
/dev/sda2 on /srv type btrfs (rw,relatime,space_cache,subvolid=264,subvol=@/srv)
/dev/sda2 on /boot/grub2/i386-pc type btrfs (rw,relatime,space_cache,subvolid=260,subvol=@/boot/grub2/i386-pc)
/dev/sda2 on /boot/grub2/x86_64-efi type btrfs (rw,relatime,space_cache,subvolid=261,subvol=@/boot/grub2/x86_64-efi)
/dev/sda2 on /home type btrfs (rw,relatime,space_cache,subvolid=262,subvol=@/home)
```

```
/dev/sda2 on /var/log type btrfs (rw,relatime,space_cache,subvolid=274,subvol=@/var/log)
/dev/mapper/vg0-lv1 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 :

```
SUSE12SP1:~ # btrfs subvolume create /mnt/volume1
Create subvolume '/mnt/volume1'
SUSE12SP1:~ # btrfs subvolume list /mnt/
ID 256 gen 6 top level 5 path volume1
SUSE12SP1:~ # ls -l /mnt
total 0
drwxr-xr-x 1 root root 0 Oct 11 16:08 volume1
```



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

LAB #23 - Convertir un Système de Fichiers Ext4 en Btrfs

Agrandissez le volume logique **/dev/vg0/lv2** :

```
SUSE12SP1:~ # lvextend -L +600M /dev/vg0/lv2
Using stripesize of last segment 64.00 KiB
Rounding size (89 extents) up to stripe boundary size for segment (90 extents)
Size of logical volume vg0/lv2 changed from 112.00 MiB (14 extents) to 720.00 MiB (90 extents).
Logical volume lv2 successfully resized
```

Créez un système de fichiers Ext4 sur **/dev/vg0/lv2** :

```
SUSE12SP1:~ # mkfs.ext4 /dev/vg0/lv2
mke2fs 1.42.11 (09-Jul-2014)
```

```
Creating filesystem with 184320 4k blocks and 46080 inodes
Filesystem UUID: 39528cf1-688e-4d72-a774-9a64039ab157
Superblock backups stored on blocks:
    32768, 98304, 163840
```

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

Convertissez maintenant le système de fichiers Ext4 en Btrfs :

```
SUSE12SP1:~ # btrfs-convert /dev/vg0/lv2
create btrfs filesystem:
    blocksize: 4096
    nodesize: 16384
    features:  extref, skinny-metadata (default)
creating btrfs metadata.
copy inodes [0] [          0/          11]
creating ext2 image file.
cleaning up system chunk.
conversion complete.
```

Visualisez les systèmes de fichiers Btrfs :

```
SUSE12SP1:~ # btrfs filesystem show
Label: none  uuid: 65337196-2d6b-4c8b-b917-30c3867bf265
    Total devices 1 FS bytes used 5.94GiB
    devid    1 size 17.99GiB used 7.32GiB path /dev/sda2

Label: none  uuid: 43e97638-f309-4769-a47a-b5d557f361f8
    Total devices 1 FS bytes used 36.00KiB
    devid    1 size 104.00MiB used 12.00MiB path /dev/mapper/vg0-lv1
```

```
Label: none  uuid: 1d2c92a7-249c-4500-9dec-6c7bbc417401
Total devices 1 FS bytes used 28.21MiB
devid    1 size 720.00MiB used 720.00MiB path /dev/mapper/vg0-lv2
```

```
btrfs-progs v4.1.2+20151002
```

Ajouter la partition **/dev/vg0/lv2** à /mnt :

```
SUSE12SP1:~ # btrfs device add /dev/vg0/lv2 /mnt/
/dev/vg0/lv2 appears to contain an existing filesystem (btrfs).
Use the -f option to force overwrite.
SUSE12SP1:~ #
SUSE12SP1:~ # btrfs device add -f /dev/vg0/lv2 /mnt/
SUSE12SP1:~ #
SUSE12SP1:~ # btrfs filesystem show
Label: none  uuid: 65337196-2d6b-4c8b-b917-30c3867bf265
Total devices 1 FS bytes used 5.94GiB
devid    1 size 17.99GiB used 7.32GiB path /dev/sda2

Label: none  uuid: 43e97638-f309-4769-a47a-b5d557f361f8
Total devices 2 FS bytes used 36.00KiB
devid    1 size 104.00MiB used 12.00MiB path /dev/mapper/vg0-lv1
devid    2 size 720.00MiB used 0.00B path /dev/mapper/vg0-lv2
```

```
btrfs-progs v4.1.2+20151002
```

```
SUSE12SP1:~ #
SUSE12SP1:~ # df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2       18G   6.5G   12G   37% /
devtmpfs        815M   8.0K   815M   1% /dev
tmpfs           824M   144K   824M   1% /dev/shm
tmpfs           824M   9.7M   814M   2% /run
tmpfs           824M    0     824M   0% /sys/fs/cgroup
/dev/sda2       18G   6.5G   12G   37% /.snapshots
```

```
/dev/sda2      18G  6.5G  12G  37% /var/spool
/dev/sda2      18G  6.5G  12G  37% /var/opt
/dev/sda2      18G  6.5G  12G  37% /var/lib/named
/dev/sda2      18G  6.5G  12G  37% /var/lib/pgsql
/dev/sda2      18G  6.5G  12G  37% /var/tmp
/dev/sda2      18G  6.5G  12G  37% /var/lib/mariadb
/dev/sda2      18G  6.5G  12G  37% /var/lib/mysql
/dev/sda2      18G  6.5G  12G  37% /var/lib/mailman
/dev/sda2      18G  6.5G  12G  37% /usr/local
/dev/sda2      18G  6.5G  12G  37% /var/crash
/dev/sda2      18G  6.5G  12G  37% /var/lib/libvirt/images
/dev/sda2      18G  6.5G  12G  37% /tmp
/dev/sda2      18G  6.5G  12G  37% /opt
/dev/sda2      18G  6.5G  12G  37% /srv
/dev/sda2      18G  6.5G  12G  37% /boot/grub2/i386-pc
/dev/sda2      18G  6.5G  12G  37% /boot/grub2/x86_64-efi
/dev/sda2      18G  6.5G  12G  37% /home
/dev/sda2      18G  6.5G  12G  37% /var/log
/dev/mapper/vg0-lv1 824M  4.1M  819M   1% /mnt
```

LAB #24 - Créer un Snapshot

Créez un snapshot du volume /mnt :

```
SUSE12SP1:~ # btrfs subvolume snapshot /mnt /mnt/snapshot
Create a snapshot of '/mnt' in '/mnt/snapshot'
SUSE12SP1:~ #
SUSE12SP1:~ # ls -l /mnt
total 4
drwxr-xr-x 1 root root 14 Oct 11 16:06 snapshot
drwxr-xr-x 1 root root  0 Oct 11 16:08 volume1
```

Démontez le système de fichiers btrfs :

```
SUSE12SP1:~ # umount /mnt
```

Montez ensuite le snapshot :

```
SUSE12SP1:~ # mount -o subvol=snapshot /dev/vg0/lv1 /mnt/  
SUSE12SP1:~ # ls -l /mnt  
total 0  
drwxr-xr-x 1 root root 0 Oct 11 16:18 volume1
```

LAB #25 - Ajouter une Etiquette au Système de Fichiers Btrfs

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

```
SUSE12SP1:~ # btrfs-show-super /dev/vg0/lv1  
superblock: bytenr=65536, device=/dev/vg0/lv1  
-----  
csum          0xd2aa2507 [match]  
bytenr        65536  
flags         0x1  
              ( WRITTEN )  
magic         _BHRfS_M [match]  
fsid          43e97638-f309-4769-a47a-b5d557f361f8  
label  
generation    11  
root          4222976  
sys_array_size 97  
chunk_root_generation 8  
root_level    1  
chunk_root    135168  
chunk_root_level 0  
log_root      0  
log_root_transid 0
```



```
log_root_level      0
total_bytes         864026624
bytes_used          49152
sectorsize          4096
nodesize            4096
leafsize            4096
stripesize          4096
root_dir            6
num_devices         2
compat_flags        0x0
compat_ro_flags     0x0
incompat_flags      0x145
                    ( MIXED_BACKREF |
                      MIXED_GROUPS |
                      EXTENDED_IREF |
                      SKINNY_METADATA )
csum_type           0
csum_size           4
cache_generation    11
uuid_tree_generation 11
dev_item.uuid       ebfacc21-4ce5-4708-8410-38dd8bed3160
dev_item.fsid       43e97638-f309-4769-a47a-b5d557f361f8 [match]
dev_item.type       0
dev_item.total_bytes 109051904
dev_item.bytes_used 12582912
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 une étiquette avec la commande **btrfs filesystem label** :

```
SUSE12SP1:~ # btrfs filesystem label /mnt my_btrfs
SUSE12SP1:~ # btrfs-show-super /dev/vg0/lv1
superblock: bytenr=65536, device=/dev/vg0/lv1
-----
csum            0x88d0a74b [match]
bytenr          65536
flags           0x1
                ( WRITTEN )
magic           _BHRfS_M [match]
fsid            43e97638-f309-4769-a47a-b5d557f361f8
label           my_btrfs
generation      12
root            4198400
sys_array_size  97
chunk_root_generation  8
root_level      1
chunk_root      135168
chunk_root_level  0
log_root        0
log_root_transid  0
log_root_level  0
total_bytes     864026624
bytes_used      49152
sectorsize     4096
nodesize       4096
leafsize       4096
stripesize     4096
root_dir        6
num_devices     2
compat_flags    0x0
compat_ro_flags 0x0
incompat_flags  0x145
```

```

( MIXED_BACKREF |
  MIXED_GROUPS |
  EXTENDED_IREF |
  SKINNY_METADATA )
csum_type      0
csum_size      4
cache_generation 12
uuid_tree_generation 12
dev_item.uuid   ebfacc21-4ce5-4708-8410-38dd8bed3160
dev_item.fsid   43e97638-f309-4769-a47a-b5d557f361f8 [match]
dev_item.type   0
dev_item.total_bytes 109051904
dev_item.bytes_used 12582912
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

```

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

Description	Ext3	Ext4	XFS	ReiserFS	JFS	Btrfs
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

LAB #26 - Créer un Système de Fichiers ISO

Pour créer un fichier ISO à partir d'une arborescence de fichiers, il convient d'utiliser la commande **mkisofs** :

```
SUSE12SP1:~ # cd /tmp
SUSE12SP1:/tmp # genisoimage -o tmp.iso /tmp
I: -input-charset not specified, using utf-8 (detected in locale settings)
Using _JAVA000.;1 for /.java_pid8404 (.java_pid1671)
Using SYSTE000.SER for /systemd-private-dc647e4673bf4bc9af99403caefb8c16-rtkit-daemon.service-k0MzkG (systemd-private-47979e849a3644dc81758975d6a508d4-rtkit-daemon.service-APKqKJ)
Using SYSTE001.SER for /systemd-private-47979e849a3644dc81758975d6a508d4-rtkit-daemon.service-APKqKJ (systemd-private-fcfd6105e7604dc9a85456df8d37f40b-rtkit-daemon.service-9mcSak)
Using _JAVA001.;1 for /.java_pid1671 (.java_pid1612)
Using _JAVA002.;1 for /.java_pid1612 (.java_pid1690)
Using _JAVA003.;1 for /.java_pid1690 (.java_pid1662)
Using _JAVA004.;1 for /.java_pid1662 (.java_pid1532)
Using SYSTE002.SER for /systemd-private-fcfd6105e7604dc9a85456df8d37f40b-rtkit-daemon.service-9mcSak (systemd-private-a9ca64d7289f4871b7c434cdad64e9b0-rtkit-daemon.service-bev5Nn)
Using _JAVA005.;1 for /.java_pid1532 (.java_pid2392)
Using _JAVA006.;1 for /.java_pid2392 (.java_pid3281)
Using MANAG000.;1 for /managera1411267841657715235client (managera5050357016347721452server)
Using MANAG001.;1 for /managera5050357016347721452server (managera4847938942232964844client)
Using MANAG002.;1 for /managera4847938942232964844client (managera3336001029897679475server)
Using SYSTE003.SER for /systemd-private-a9ca64d7289f4871b7c434cdad64e9b0-rtkit-daemon.service-bev5Nn (systemd-private-04f820fa26c745be8ddb814c6292f21-rtkit-daemon.service-o4lKP5)
Using RUN_C000.283;1 for /tmp/run-crons.tGyimF/run-crons.daily.2832 (run-crons_output.2832)
```

```
Using RUN_C001.283;1 for /tmp/run-crons.tGyimF/run-crons_output.2832 (run-crons_mail.2832)
Total translation table size: 0
Total rockridge attributes bytes: 0
Total directory bytes: 51200
Path table size(bytes): 414
Max brk space used 1a000
713 extents written (1 MB)
```

Le fichier ISO peut être monter en utilisant un périphérique de type **loop** :

```
SUSE12SP1:/tmp # mount tmp.iso /mnt -o loop
mount: /dev/loop0 is write-protected, mounting read-only
SUSE12SP1:/tmp # ls /mnt
_esd_100  _ice_uni  _java002  _java005  _test_un  _xim_uni  manag000  managera  repertoï  syste001.ser
systemd_.ser  tmp.iso
_esd_486  _java000  _java003  _java006  _x0_lock  hsperfda  manag001  mc_root  run_cron.tgy  syste002.ser
systemic  vboxgues.sym
_font_un  _java001  _java004  _java_pi  _x11_uni  inherit  manag002  public_d  syste000.ser  syste003.ser
systemic  vg0_back
```

Démontez maintenant le fichier ISO :

```
SUSE12SP1:/tmp # cd ~
SUSE12SP1:~ # umount /tmp/tmp.iso
```

Options de la Commande mkisofs

Les options de cette commande sont :

```
SUSE12SP1:~ # genisoimage --help
Usage: genisoimage [options] file...
Options:
```

```
-nobak                Do not include backup files
-no-bak              Do not include backup files
-abstract FILE       Set Abstract filename
-A ID, -appid ID     Set Application ID
-biblio FILE         Set Bibliographic filename
-cache-inodes        Cache inodes (needed to detect hard links)
-no-cache-inodes     Do not cache inodes (if filesystem has no unique unides)
-check-oldnames      Check all imported ISO9660 names from old session
-check-session FILE  Check all ISO9660 names from previous session
-copyright FILE      Set Copyright filename
-debug              Set debug flag
-b FILE, -eltorito-boot FILE
                    Set El Torito boot image name
-eltorito-alt-boot   Start specifying alternative El Torito boot parameters
-B FILES, -sparc-boot FILES Set sparc boot image names
-sunx86-boot FILES   Set sunx86 boot image names
-G FILE, -generic-boot FILE Set generic boot image name
-sparc-label label text Set sparc boot disk label
-sunx86-label label text Set sunx86 boot disk label
-c FILE, -eltorito-catalog FILE
                    Set El Torito boot catalog name
-C PARAMS, -cdrecord-params PARAMS
                    Magic paramters from cdrecord
-d, -omit-period     Omit trailing periods from filenames (violates ISO9660)
-dir-mode mode       Make the mode of all directories this mode.
-D, -disable-deep-relocation
                    Disable deep directory relocation (violates ISO9660)
-file-mode mode      Make the mode of all plain files this mode.
-f, -follow-links    Follow symbolic links
-gid gid             Make the group owner of all files this gid.
-graft-points        Allow to use graft points for filenames
-root DIR            Set root directory for all new files and directories
-old-root DIR        Set root directory in previous session that is searched for files
-help               Print option help
```

```
-hide GLOBFILE          Hide ISO9660/RR file
-hide-list FILE         File with list of ISO9660/RR files to hide
-hidden GLOBFILE        Set hidden attribute on ISO9660 file
-hidden-list FILE       File with list of ISO9660 files with hidden attribute
-hide-joliet GLOBFILE   Hide Joliet file
-hide-joliet-list FILE  File with list of Joliet files to hide
-hide-joliet-trans-tbl  Hide TRANS.TBL from Joliet tree
-hide-rr-moved          Rename RR_MOVED to .rr_moved in Rock Ridge tree
-gui                    Switch behaviour for GUI
-i ADD_FILES            No longer supported
-input-charset CHARSET  Local input charset for file name conversion
-output-charset CHARSET Output charset for file name conversion
-iso-level LEVEL        Set ISO9660 conformance level (1..3) or 4 for ISO9660 version 2
-J, -joliet             Generate Joliet directory information
-joliet-long            Allow Joliet file names to be 103 Unicode characters
-jcharset CHARSET       Local charset for Joliet directory information
-l, -full-iso9660-filenames Allow full 31 character filenames for ISO9660 names
-max-iso9660-filenames  Allow 37 character filenames for ISO9660 names (violates ISO9660)
-allow-limited-size     Allow different file sizes in ISO9660/UDF on large files
-allow-leading-dots     Allow ISO9660 filenames to start with '.' (violates ISO9660)
-ldots                  Allow ISO9660 filenames to start with '.' (violates ISO9660)
-L, -allow-leading-dots Allow ISO9660 filenames to start with '.' (violates ISO9660)
-log-file LOG_FILE      Re-direct messages to LOG_FILE
-m GLOBFILE, -exclude GLOBFILE
                        Exclude file name
-exclude-list FILE      File with list of file names to exclude
-pad                    Pad output to a multiple of 32k (default)
-no-pad                 Do not pad output to a multiple of 32k
-M FILE, -prev-session FILE Set path to previous session to merge
-dev SCSIdev           Set path to previous session to merge
-N, -omit-version-number Omit version number from ISO9660 filename (violates ISO9660)
-new-dir-mode mode      Mode used when creating new directories.
-force-rr              Inhibit automatic Rock Ridge detection for previous session
-no-rr                 Inhibit reading of Rock Ridge attributes from previous session
```

```
-no-split-symlink-components      Inhibit splitting symlink components
-no-split-symlink-fields          Inhibit splitting symlink fields
-o FILE, -output FILE            Set output file name
-path-list FILE                  File with list of pathnames to process
-p PREP, -preparer PREP         Set Volume preparer
-print-size                      Print estimated filesystem size and exit
-publisher PUB                  Set Volume publisher
-P PUB, -publisher PUB          Set Volume publisher
-quiet                           Run quietly
-r, -rational-rock              Generate rationalized Rock Ridge directory information
-R, -rock                        Generate Rock Ridge directory information
-s TYPE, -sectype TYPE          Set output sector type to e.g. data/xal/raw
-alpha-boot FILE                Set alpha boot image name (relative to image root)
-hppa-cmdline CMDLINE           Set hppa boot command line (relative to image root)
-hppa-kernel-32 FILE            Set hppa 32-bit image name (relative to image root)
-hppa-kernel-64 FILE            Set hppa 64-bit image name (relative to image root)
-hppa-bootloader FILE           Set hppa boot loader file name (relative to image root)
-hppa-ramdisk FILE              Set hppa ramdisk file name (relative to image root)
-mips-boot FILE                 Set mips boot image name (relative to image root)
-mipsel-boot FILE               Set mipsel boot image name (relative to image root)
-jigdo-jigdo FILE               Produce a jigdo .jigdo file as well as the .iso
-jigdo-template FILE            Produce a jigdo .template file as well as the .iso
-jigdo-min-file-size SIZE       Minimum size for a file to be listed in the jigdo file
-jigdo-force-md5 PATTERN        Pattern(s) where files MUST match an externally-supplied MD5sum
-jigdo-exclude PATTERN          Pattern(s) to exclude from the jigdo file
-jigdo-map PATTERN1=PATTERN2    Pattern(s) to map paths (e.g. Debian=/mirror/debian)
-md5-list FILE                  File containing MD5 sums of the files that should be checked
-jigdo-template-compress ALGORITHM
                                Choose to use gzip or bzip2 compression for template data; default is gzip
-checksum_algorithm_iso alg1,alg2,...
                                Specify the checksum types desired for the output image
-checksum_algorithm_template alg1,alg2,...
```



```
Specify the checksum types desired for the output jigdo template
-sort FILE          Sort file content locations according to rules in FILE
-split-output       Split output into files of approx. 1GB size
-stream-file-name FILE_NAME Set the stream file ISO9660 name (incl. version)
-stream-media-size # Set the size of your CD media in sectors
-sysid ID           Set System ID
-T, -translation-table Generate translation tables for systems that don't understand long filenames
-table-name TABLE_NAME Translation table file name
-ucs-level LEVEL    Set Joliet UCS level (1..3)
-udf                Generate UDF file system
-dvd-video          Generate DVD-Video compliant UDF file system
-uid uid            Make the owner of all files this uid.
-U, -untranslated-filenames Allow Untranslated filenames (for HPUX & AIX - violates ISO9660). Forces -l, -d, -N, -allow-leading-dots, -relaxed-filenames, -allow-lowercase, -allow-multidot
-relaxed-filenames Allow 7 bit ASCII except lower case characters (violates ISO9660)
-no-iso-translate   Do not translate illegal ISO characters '~', '-' and '#' (violates ISO9660)
-allow-lowercase    Allow lower case characters in addition to the current character set (violates ISO9660)
-allow-multidot     Allow more than one dot in filenames (e.g. .tar.gz) (violates ISO9660)
-use-fileversion LEVEL Use file version # from filesystem
-v, -verbose        Verbose
-version            Print the current version
-V ID, -volid ID    Set Volume ID
-volset ID          Set Volume set ID
-volset-size #      Set Volume set size
-volset-seqno #     Set Volume set sequence number
-x FILE, -old-exclude FILE Exclude file name(depreciated)
-hard-disk-boot     Boot image is a hard disk image
-no-emul-boot       Boot image is 'no emulation' image
-no-boot            Boot image is not bootable
-boot-load-seg #    Set load segment for boot image
-boot-load-size #   Set numbers of load sectors
-boot-info-table    Patch boot image with info table
-XA                Generate XA directory attributes
```

```
-xa                Generate rationalized XA directory attributes
-z, -transparent-compression
                    Enable transparent compression of files
-hfs-type TYPE     Set HFS default TYPE
-hfs-creator CREATOR Set HFS default CREATOR
-g, -apple        Add Apple IS09660 extensions
-h, -hfs          Create IS09660/HFS hybrid
-map MAPPING_FILE Map file extensions to HFS TYPE/CREATOR
-H MAPPING_FILE, -map MAPPING_FILE
                    Map file extensions to HFS TYPE/CREATOR
-magic FILE       Magic file for HFS TYPE/CREATOR
-probe            Probe all files for Apple/Unix file types
-mac-name         Use Macintosh name for IS09660/Joliet/RockRidge file name
-no-mac-files     Do not look for Unix/Mac files (deprecated)
-boot-hfs-file FILE Set HFS boot image name
-part            Generate HFS partition table
-cluster-size SIZE Cluster size for PC Exchange Macintosh files
-auto FILE       Set HFS AutoStart file name
-no-desktop       Do not create the HFS (empty) Desktop files
-hide-hfs GLOBFILE Hide HFS file
-hide-hfs-list FILE List of HFS files to hide
-hfs-volid HFS_VOLID Volume name for the HFS partition
-icon-position    Keep HFS icon position
-root-info FILE  finderinfo for root folder
-input-hfs-charset CHARSET Local input charset for HFS file name conversion
-output-hfs-charset CHARSET Output charset for HFS file name conversion
-hfs-unlock      Leave HFS Volume unlocked
-hfs-bless FOLDER_NAME Name of Folder to be blessed
-hfs-parms PARAMETERS Comma separated list of HFS parameters
-prep-boot FILE  PReP boot image file -- up to 4 are allowed
-chrp-boot       Add CHRP boot header
--cap            Look for AUFS CAP Macintosh files
--netatalk       Look for NETATALK Macintosh files
--double         Look for AppleDouble Macintosh files
```

```
--ethershare      Look for Helios EtherShare Macintosh files
--exchange        Look for PC Exchange Macintosh files
--sgi             Look for SGI Macintosh files
--macbin          Look for MacBinary Macintosh files
--single          Look for AppleSingle Macintosh files
--ushare          Look for IPT UShare Macintosh files
--xinet           Look for XINET Macintosh files
--dave            Look for DAVE Macintosh files
--sfm             Look for SFM Macintosh files
--osx-double      Look for MacOS X AppleDouble Macintosh files
--osx-hfs         Look for MacOS X HFS Macintosh files
```

Report problems to debburn-devel@lists.alioth.debian.org.

Systemes de Fichiers Chiffrés

LAB #27 - Créer un Système de Fichiers Chiffré avec encryptfs

Le chiffrement dans ce cas consiste à monter un système de fichiers sur lui-même. Le verrouillage/déverrouillage se fait par une de trois méthodes :

- Phrase de passe,
- tspi - une interface de chiffrement vers d'autres programmes,
- openssl.

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

```
SUSE12SP1:~ # zypper install ecryptfs-utils
Refreshing service 'SUSE_Linux_Enterprise_Server_12_SP1_x86_64'.
Loading repository data...
Reading installed packages...
Resolving package dependencies...
```

The following 14 NEW packages are going to be installed:

```
ecryptfs-utils ecryptfs-utils-32bit gnome-keyring-32bit libgmp10-32bit libgnutls28-32bit libhogweed2-32bit  
libnettle4-32bit libp11-kit0-32bit  
libpkcs11-helper1 libpkcs11-helper1-32bit libtasn1-6-32bit libtspi1 libtspi1-32bit trousers
```

14 new packages to install.

Overall download size: 2.2 MiB. Already cached: 0 B. After the operation, additional 5.2 MiB will be used.

Continue? [y/n/? shows all options] (y): y

Démontez le système de fichiers btrfs sur /mnt :

```
SUSE12SP1:~ # umount /mnt  
SUSE12SP1:~ # ls /mnt  
lvm sdb8 sdb9 testsnap
```

Montez maintenant /mnt/sdb8 sur lui-même :

```
SUSE12SP1:~ # mount -t ecryptfs /mnt/sdb8 /mnt/sdb8  
Select key type to use for newly created files:  
 1) openssl  
 2) passphrase  
 3) pkcs11-helper  
 4) tspi  
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]:
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** :

```
SUSE12SP1:~ # df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2       18G   6.6G   11G   38% /
devtmpfs        815M   8.0K   815M    1% /dev
tmpfs           824M  144K   824M    1% /dev/shm
tmpfs           824M   9.8M   814M    2% /run
tmpfs           824M     0   824M    0% /sys/fs/cgroup
```

```
/dev/sda2      18G  6.6G  11G  38% /.snapshots
/dev/sda2      18G  6.6G  11G  38% /var/spool
/dev/sda2      18G  6.6G  11G  38% /var/opt
/dev/sda2      18G  6.6G  11G  38% /var/lib/named
/dev/sda2      18G  6.6G  11G  38% /var/lib/pgsql
/dev/sda2      18G  6.6G  11G  38% /var/tmp
/dev/sda2      18G  6.6G  11G  38% /var/lib/mariadb
/dev/sda2      18G  6.6G  11G  38% /var/lib/mysql
/dev/sda2      18G  6.6G  11G  38% /var/lib/mailman
/dev/sda2      18G  6.6G  11G  38% /usr/local
/dev/sda2      18G  6.6G  11G  38% /var/crash
/dev/sda2      18G  6.6G  11G  38% /var/lib/libvirt/images
/dev/sda2      18G  6.6G  11G  38% /tmp
/dev/sda2      18G  6.6G  11G  38% /opt
/dev/sda2      18G  6.6G  11G  38% /srv
/dev/sda2      18G  6.6G  11G  38% /boot/grub2/i386-pc
/dev/sda2      18G  6.6G  11G  38% /boot/grub2/x86_64-efi
/dev/sda2      18G  6.6G  11G  38% /home
/dev/sda2      18G  6.6G  11G  38% /var/log
/mnt/sdb8      477M  2.3M  445M   1% /mnt/sdb8
```

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

```
SUSE12SP1:~ # mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
...
/dev/sdb8 on /mnt/sdb8 type ext4 (rw,relatime,data=ordered)
/mnt/sdb8 on /mnt/sdb8 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/sdb8 :

```
SUSE12SP1:~ # touch /mnt/sdb8/encrypt
```

```
SUSE12SP1:~ # echo "fenestros" > /mnt/sdb8/encrypt
SUSE12SP1:~ # cat /mnt/sdb8/encrypt
fenestros
```

Démontez maintenant /mnt/sdb8 :

```
SUSE12SP1:~ # umount /mnt/sdb8
SUSE12SP1:~ # mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
...
/dev/sdb8 on /mnt/sdb8 type ext4 (rw,relatime,data=ordered)
```



Important : Notez que /dev/sdb8 est toujours monté sur /mnt/sdb8.

Constatez maintenant le contenu de /mnt/sdb8 :

```
SUSE12SP1:~ # ls -l /mnt/sdb8
total 24
-rw-r--r-- 1 root root 12288 Oct 12 11:25
ECRYPTFS_FNEK_ENCRYPTED.FWaFfjrdapd7RkRCT3OFIdaE.-6mxquDYm8R4p7VFuV0SGgSJauuQJ2hEE--
drwx----- 2 root root 12288 Oct 11 13:42 lost+found
```



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

Constatez maintenant le contenu du fichier

ECRYPTFS_FNEK_ENCRYPTED.FWaFfjrdapd7RkRCT3OFIdaE.-6mxquDYm8R4p7VFuV0SGgSJauuQJ2hEE- :

```
SUSE12SP1:~ # cat /mnt/sdb8/ECRYPTFS_FNEK_ENCRYPTED.FWaFfjrdapd7RkRCT3OFIdaE.-6mxquDYm8R4p7VFuV0SGgSJauuQJ2hEE--
```

```

0ND0000
0000000"3DUfw`
R0v000U`000~0000_CONSOLE00000ZIw=0A000'0000m00P000]0z0000]Zt\9>000!E0000050000g0`0$sh000^809"0200_{00\00'000000:
Y0Q000Q?Y000=v000/00/c00Q800T;=0K0.0)S0000C0'&0"B8n00"000000g0k[0m00YA0_0000C0u(xo
x,]81dту&4t0u000B000ID00V0D0E00000zj006/t0000|00/000`](0^0V0t0|00R0UI0z%00000000Z00M0(\`0*7s000,0B0\H*00
>0000#04i000S3S00000I0000$Z!00U0B000700`0a000hT y0D0000;000W00.0000&00~00a0|0000U00UG
Eg0000p0?0V
=D0l:0_0600}C0·d0G0v00000$R[(0X400;0?0g{0eh00020A
0000z3.b0EP9160'000000'dFc0000t000076W0r8^0
00>c0s0Y0sZ00!0lt
e0`000D00)00R^10zM0W2000s00]hc0T) Y0000xm001y0+.00<0B00Tn\00h&00;y(0Nz0z0j0^00u000000000)0CH0g0tN00|G00
y000000H0QC000S0;C#$`,X000'I000089x00/0000e0f)QI0000fnv0$!0000GV000060類
TJ00000;00000000sè000x0q0
Xk00+A00T@?0M0Vo0000iVR0000800`#N0H}0F0)0000j_0*00Ä0R0%0"i300sx(0wn01o000000h0;h2@0010000@0s9}00r|00K00{00000Ft0
V000~00f0%0
000000w00x0fu0>0(
0}r"
00F00Dv00;0000 003(00000Vc/k000F}0
0000000kXb0P00S0Q'000mA^(00D<]02
0.000Ta-#00i00J40000-
00$00Pu0Z00#0S000000`00zb00R0008&N"f0'000000Ux0.0"0{80000y0T0Hs
p000{N00 0E?38Y00#t0o0000q00H`01000000<0000000LLU6B0.^wX0)00W00000=0000{000Zn0000()d000000000
40&*00m00050003[btPA000E060g0mg00000300<00000/0!0j00 0kR009{00000t0YQ0%C0n00?B000^0000zVD00 ]0
00b0000000}0j00$q6u005000.
0000 00Moz0000/000000G!'y0F
0H00i07000i0:<0=<000sp00L0g00=b0$400J0p000-0000(000D%0z000H000000l08<0CM0000t00000a0Y0000000+00}0bV0Gc"0000200?f
c000c`.X00.4`00000c02e0A0000n100
0000y00RU*000y0=0k0G000[SXKM`000e00d.000K^F0+0N000~0Q00D0?000ud00M00Wxu}Ks000J00V~0a00?00A007J00000h!00N00000x0
000.
!0@03v:0X0B$0V0_E00000J0000

```



```

    *G0v00jV00*0?000400*0â00Kπ0v=)00{`00000bEQkð0900xh)0&7_700v00zL"09H0
[rA]m_oT30S00m0/0v9000w00J00B^0^0U0"0c'}0000-e00V0r0 B
y00uE000g100f[[0(0R0uZ000YPz($Ldc00*P).000008$0g<0TFlvDc002I
    H[o050~S000

0000.0b0^00%00<0
P00KQ00;#%00N+C0W0S0zZe00000[0d00000/0t|00
P,00000 00%00%
    000000`00s0|0
    0d0Y<0000F0000L.0000rWN0B00}00000200A0107Ua&000
    麋0a0s00000000Y0IO
`?(0f@):0;/0000^0030V00D000W`04"0000000gTK000!Yw0A00P=Z0.P<,0%W0000    00 0;0i00U^0%0f0F000s000`050I-
/$G00]zj0Z00"0006d0*:04i0W00]
'00000w0D$e`00"08000040}&00B0fé0001000o00;00@000000=0?`xZn00+40000000
0000x0w0000X`0\w00F00l000Jmk0,0M0
    NyNH00kA00h000b00e0v]0i00z0j0000't0K,F0>0000$r-x0i05000900E000>/00Z00e000G00000Rx00H,0c"0000
0<0,*0000000^02n0g}G    l0B0iBq0000CB000000000L
90;0_0q
0k00>0Hzj0o0SM000i0x60000q0 000&000000000^L0r00P0x0뽕000000Bu0}000(z0008?0U000-0B000F}000
0#j0000000067i00V2L00000$0P4X0;0~0
0?0i0B000000060Ëa&Ed08@000o'0S00p0^D00    000u200[0I0a0(1000M{pA0_x)0a00=0(00{:n000l0o0
0<P,00h0z00DH000000C7Z000$00pT005Lw0k0{0000{

200#0B0000300100200(00@000020[000000W00000.0000_M0Y000p-0?00/n0L0l0~0A0 |0R000Gj%00j00-N0000t0,8}0Y<E080
0000;00000K0t0
000nfi000m000%qW0qC00000000<hX000$00020yB00z000d0;0)000`E00    k0P~0000E00,IC
0xEk0FHePd000000|00000000B0GH00000e0z050+,/0vB000n00x}0%0
`m0o-50NU00.000vU000e<000]000VV0Z0500tF#C000000000+n0000yr0A0M0L0/hl00%#K20!000>0'000%0?0000@000
o%R
R00y0]0e5007:00B10Yp000ym00r000I2Ny0
    0B00m0$zI0I0    @0Hrt0U00P0x*00r00]00>CMq000<|00    0000000000h0H!00-
00+"00I0000x02fJP1300@-030Z1D0>0N0 0f00t0-

```




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

Initialisez la partition avec LUKS :

```
SUSE12SP1:~ # cryptsetup --verbose --verify-passphrase luksFormat /dev/sdb9
```

```
WARNING!
```

```
=====
```

```
This will overwrite data on /dev/sdb9 irrevocably.
```

```
Are you sure? (Type uppercase yes): YES
```

```
Enter passphrase: fenestros123456789
```

```
Verify passphrase: fenestros123456789
```

```
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 **sdb9** :

```
SUSE12SP1:~ # cryptsetup luksOpen /dev/sdb9 sdb9  
Enter passphrase for /dev/sdb9:fenestros123456789
```



Important : La passphrase ne sera pas en claire.

Vérifiez que le système voit la partition :

```
SUSE12SP1:~ # ls -l /dev/mapper | grep sdb9
lrwxrwxrwx 1 root root      7 Oct 12 11:38 sdb9 -> ../dm-2
```

Créez maintenant un système de fichiers sur **/dev/mapper/sdb9** :

```
SUSE12SP1:~ # mkfs.ext4 /dev/mapper/sdb9
mke2fs 1.42.11 (09-Jul-2014)
Creating filesystem with 202752 1k blocks and 50800 inodes
Filesystem UUID: a37da1b7-7381-49e4-9c99-819671b12ebc
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 la partition LUKS :

```
SUSE12SP1:~ # mount /dev/mapper/sdb9 /mnt/sdb9
```

Vérifiez la présence du montage :

```
SUSE12SP1:~ # df -h | grep sdb9
/dev/mapper/sdb9 188M 1.6M 173M 1% /mnt/sdb9
```

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

```
SUSE12SP1:~ # vi /etc/crypttab
SUSE12SP1:~ # cat /etc/crypttab
sdb9    /dev/sdb9    none
```

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

```
SUSE12SP1:~ # vi /etc/fstab
SUSE12SP1:~ # cat /etc/fstab
UUID=9e0e4497-5e47-419a-8ff4-9e2dd7d6a11b swap swap defaults 0 0
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 / btrfs defaults 0 0
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /boot/grub2/i386-pc btrfs subvol=@/boot/grub2/i386-pc 0 0
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /boot/grub2/x86_64-efi btrfs subvol=@/boot/grub2/x86_64-efi 0 0
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /home btrfs subvol=@/home 0 0
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /opt btrfs subvol=@/opt 0 0
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /srv btrfs subvol=@/srv 0 0
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /tmp btrfs subvol=@/tmp 0 0
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /usr/local btrfs subvol=@/usr/local 0 0
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /var/crash btrfs subvol=@/var/crash 0 0
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /var/lib/libvirt/images btrfs subvol=@/var/lib/libvirt/images 0 0
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /var/lib/mailman btrfs subvol=@/var/lib/mailman 0 0
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /var/lib/mariadb btrfs subvol=@/var/lib/mariadb 0 0
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /var/lib/mysql btrfs subvol=@/var/lib/mysql 0 0
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /var/lib/named btrfs subvol=@/var/lib/named 0 0
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /var/lib/pgsql btrfs subvol=@/var/lib/pgsql 0 0
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /var/log btrfs subvol=@/var/log 0 0
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /var/opt btrfs subvol=@/var/opt 0 0
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /var/spool btrfs subvol=@/var/spool 0 0
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /var/tmp btrfs subvol=@/var/tmp 0 0
UUID=65337196-2d6b-4c8b-b917-30c3867bf265 /.snapshots btrfs subvol=@/.snapshots 0 0
/dev/mapper/sdb9 /mnt/sdb9 ext4 defaults 1 2
```

Redémarrez votre machine virtuelle :

```
SUSE12SP1:~ # shutdown -r now
```



Important : Lors du démarrage de la machine virtuelle, le système devrait vous demander d'entrer la passphrase **fenestros123456789** pour permettre le montage de /dev/sdb9.

Ajouter une deuxième Passphrase

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

```
SUSE12SP1:~ # cryptsetup luksAddKey /dev/sdb9
Enter any existing passphrase: fenestros123456789
Enter new passphrase for key slot: sles123456789
Verify passphrase:
SUSE12SP1:~ #
```



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** :

```
SUSE12SP1:~ # cryptsetup luksRemoveKey /dev/sdb9
Enter passphrase to be deleted:
SUSE12SP1:~ # sles123456789
```

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**.

```
SUSE12SP1:~ # swapon -s
Filename          Type          Size      Used      Priority
/dev/sda1         partition    2103292    0         -1
```



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 partitions de swap.

Options de la Commande

Les options de la commande swapon sont :

```
SUSE12SP1:~ # swapon --help
```

Usage:

```
swapon [options] [<spec>]
```

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
-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 headings, use with --show
    --raw               use the raw output format, use with --show
    --bytes             display swap size in bytes in --show output
-v, --verbose            verbose mode

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

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. (swapon)
pages     : discard freed pages before they are reused.
* if no policy is selected both discard types are enabled. (default)
```

Available columns (for --show):

```
NAME  device file or partition path
TYPE  type of the device
SIZE  size of the swap area
USED  bytes in use
PRIO  swap priority
```

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/sda1**. Pour la désactiver, il convient de saisir la commande suivante :

```
SUSE12SP1:~ # swapoff /dev/sda1
SUSE12SP1:~ # swapon -s
SUSE12SP1:~ #
```

Options de la Commande

```
SUSE12SP1:~ # swapoff --help
```

Usage:

```
swapoff [options] [<spec>]
```

Options:

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

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

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)`.

Remettez en place votre swap :

```
SUSE12SP1:~ # swapon -p1 /dev/sda1
```

```
SUSE12SP1:~ # swapon -s
```

Filename	Type	Size	Used	Priority		
/dev/sda1		partition	2103292	0	1	

<html>

Copyright © 2004-2017 I2TCH LIMITED.

</html>

