

Comment monter un fichier image

Table of Contents

[Commande losetup](#)
[kpartx](#)

Linux permet de stocker des systèmes de fichiers dans des fichiers image et de les manipuler de manière transparente, grâce à un mécanisme de périphériques de type bloc virtuel. Le principe est d'associer le fichier à l'un de ces périphériques, et d'utiliser simplement le fichier spécial de périphérique correspondant comme un fichier spécial de périphérique de type bloc classique. Il est ainsi possible de construire une image de systèmes de fichiers de manière tout à fait transparente, et d'accéder aux données stockées dans une image de CD-ROM de manière tout aussi facile.

Les fichiers spéciaux de périphérique utilisés pour manipuler les fichiers images sont les fichiers `/dev/loopX`, où X est un numéro allant de 0 à 7. Ces fichiers ne sont utilisables que si l'option « Loopback device support » du menu « Block devices » a été activée dans la configuration du noyau.

Commande losetup

L'association entre le périphérique de type bloc virtuel et le support des données peut être réalisée à l'aide de la commande **losetup**. La syntaxe la plus simple de cette commande est la suivante :

```
losetup /dev/loopX fichier
```

où loopX est le fichier spécial de périphérique de type bloc virtuel à utiliser, et fichier est le fichier dans lequel les données de ce périphérique doivent être stockées. Une fois cette association faite, on peut utiliser ce périphérique comme un véritable périphérique. Ainsi, pour créer un système de fichiers dans un fichier image de 10 Mo, on procédera comme suit :

```
dd if=/dev/zero of=image bs=1024 count=10240  
losetup /dev/loop0 image  
mkfs -t ext2 /dev/loop0
```



La commande **dd** permet de copier un certain nombre de blocs, ici 10240 (option count) de 1024 octets (taille indiquée par l'option bs) du fichier spécial de périphérique `/dev/zero` dans le fichier image.

En réalité, les commandes de création de systèmes de fichiers peuvent également créer directement les systèmes de fichiers dans des fichiers images. La manière classique de procéder a toutefois été présentée ici pour la bonne compréhension du mécanisme de gestion des fichiers images.

Le montage d'un système de fichiers placé dans l'image disque est ensuite réalisable de manière tout

à fait classique, en utilisant le fichier spécial de périphérique `/dev/loopX` adéquat dans la commande **mount**. Toutefois, cette commande permet de réaliser le montage d'une image de manière simplifiée grâce à son option `loop`. Celle-ci permet de configurer le périphérique virtuel image de manière automatique avant de réaliser le montage. Elle prend donc en paramètre le fichier spécial de périphérique virtuel à utiliser, ainsi que le chemin sur le fichier image contenant le système de fichiers. Ainsi, pour monter une image de CD-ROM, on peut utiliser la commande suivante :

```
mount -t iso9660 -o ro,loop=/dev/loop0 image /mnt/cdrom
```

Lorsque l'on n'a plus besoin du périphérique de type bloc virtuel, il est possible de supprimer l'association avec le support de données simplement avec la commande suivante :

```
losetup -d /dev/loopX
```

où X est toujours le numéro du fichier spécial de périphérique virtuel. Lorsque l'on utilise l'option `loop` de la commande `mount`, cette opération est réalisée automatiquement par la commande `umount` et ne doit plus être effectuée manuellement.

Partitions multiples: Les images des distributions linux présentent généralement au moins deux partitions, la première pour le boot et la seconde pour le rootfs.



La commande `losetup --list` permet de lister ces devices:

```
losetup --list
NAME SIZE LIMIT OFFSET AUTOCLEAR RO BACK-FILE DIO
/dev/loop1 0 0 0 0 /home/milosz/second_loop.fs 1
/dev/loop0 0 0 0 0 /home/milosz/loop.fs 0
```

kpartx

La commande **kpartx** est fournie par le package **kpartx** sur les systèmes basés sur Debian et également sur les systèmes basés sur Red Hat, où il est généralement installé par défaut.

```
apt-get install kpartx
```



Il existe deux commandes portant le même nom pour mapper les partitions : **partx**, qui fait partie du package **util-linux**, et un dérivé nommé **kpartx**, qui fait partie de **multipath-tools**. Les principales différences sont que :

- **kpartx** peut créer des mappages pour les périphériques de bloc, tandis que **partx** est limité aux fichiers normaux.
- **partx** est plus susceptible d'avoir été installé par défaut.
- tandis que **kpartx** crée des mappages dans `/dev/mapper/`, **partx** les place dans `/dev/`



- ils acceptent des arguments similaires et ont tous deux la capacité de gérer plusieurs types de table de partition (y compris MBR et GPT).

Pour présenter les partitions en tant que périphériques de bloc séparés, utiliser l'option **-a** (ajouter) pour demander qu'un nouveau mappage soit créé pour chaque partition trouvée, et l'option **-v** (verbeux) pour obtenir une liste de ces mappages :

```
kpartx -av /dev/vg0/foo
```

En cas de succès, la sortie doit ressembler à ce qui suit :

```
add map vg0-foo1 (252:3): 0 1048576 linear 252:2 2048
add map vg0-foo2 (252:4): 0 524288 linear 252:2 1050624
add map vg0-foo3 (252:5): 0 522240 linear 252:2 1574912
```

Les mappages sont créés dans `/dev/mapper/`, donc la première partition répertoriée serait présentée comme `/dev/mapper/vg0-foo1`.

Monter le système de fichiers de la manière normale, par exemple pour monter la première partition :

```
mkdir /mnt/vmroot
mount /dev/mapper/vg0-foo1 /mnt/vmroot
```

Il est également possible d'utiliser des commandes telles que **mkfs**, **fsck**, **resize2fs**, ou toute autre chose qui fonctionne sur une partition.

Les partitions présentées à l'aide de **kpartx** ne doivent normalement pas être répertoriées dans `/etc/fstab`, car l'effet de **kpartx** ne persiste pas lors d'un redémarrage. Si un accès à long terme est requis, les commandes **kpartx** et **mount** devront être scriptées pour s'exécuter au démarrage.

Partitions multiples: Les images des distributions linux présentent généralement au moins deux partitions, la première pour le boot et la seconde pour le rootfs.



On peut afficher où les devices sont mappés avec `kpartx -lv`. Par exemple:

```
sudo kpartx -lv /dev/loop0
loop0p1 : 0 122880 /dev/loop0 8192
loop0p2 : 0 8257536 /dev/loop0 131072
```



Les commandes suivantes permettent de récupérer les id des devices avec **kpartx**:

```
LOOPDEVS=$(sudo kpartx -avs ${IMGFILE} | awk '{print $3}')
```



```
LOOPDEVB00T=/dev/mapper/$(echo ${LOOPDEVS} | awk '{print $1}')
```

```
LOOPDEVVR00TFS=/dev/mapper/$(echo ${LOOPDEVS} | awk '{print $2}')
```

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=howto:mount-img-block-device>

Last update: **2025/02/19 10:59**

