

dd sur Linux clone, sauvegarde copie de disque

Table des matières

- [1 dd sur Linux : clone, sauvegarde et copie de partition de disque](#)
 - [1.1 Cloner un disque, SSD](#)
 - [1.2 Créer une image de disque](#)
 - [1.3 Créer un fichier ISO avec dd](#)
 - [1.4 Vider le contenu d'un disque](#)
- [2 Paramètres de la commande dd](#)
 - [2.1 conv](#)
 - [2.2 bs – block size](#)
- [3 Liens](#)

dd sur Linux : clone, sauvegarde et copie de partition de disque

Voici l'utilisation basique de dd :

```
dd if=<source> of=<cible>
```

On utilise donc une source et une cible pour spécifier l'ordre de la copie.

La source et la destination peuvent être un disque dur, une partition de disque un DVD-Rom mais aussi un fichier.

- Dans le cas d'un disque, ce sera donc : /dev/sda, /dev/sdb
- Avec une partition de disque : /dev/sda1, /dev/sda2, /dev/sdb1

Suivre alors cet article :

☐ [Les partitions de disque sur Linux](#)

☐ Il faut donc bien faire attention dans la source et la destination car si vous inversez, cela écrase les données existantes. Vous risquez alors de perdre des données.

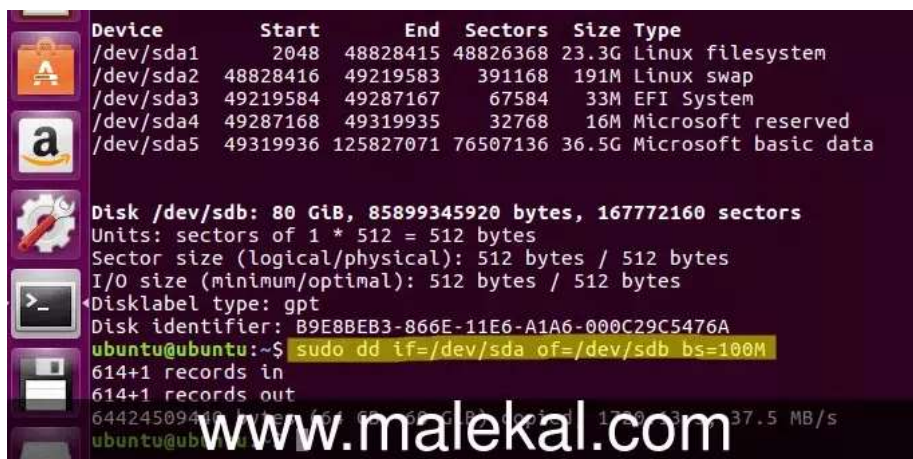
Cloner un disque, SSD

Cloner un disque dur revient à copier le contenu de ce dernier d'un disque ou SSD source vers un disque de destination.
Avec dd, c'est extrêmement simple.
Pour cloner le disque /dev/sda vers le disque /dev/sdb, on l'utilise comme ceci :

```
dd if=/dev/sda of=/dev/sdb
```

Il existe un article complet pour cloner un disque ou SSD sur Linux avec dd :

☐ [Cloner un disque \(HDD, SSD\) avec Linux](#)



Créer une image de disque

On peut ensuite **créer une image de disque**.
Cela consiste à copier l'intégralité d'un disque ou d'une partition de disque vers un fichier.
On peut ensuite restaurer le contenu des données vers un autre disque.

Par exemple pour copier tout le contenu du disque /dev/sda vers le fichier partition.img :

```
dd if=/dev/sda of=/partition.img
```

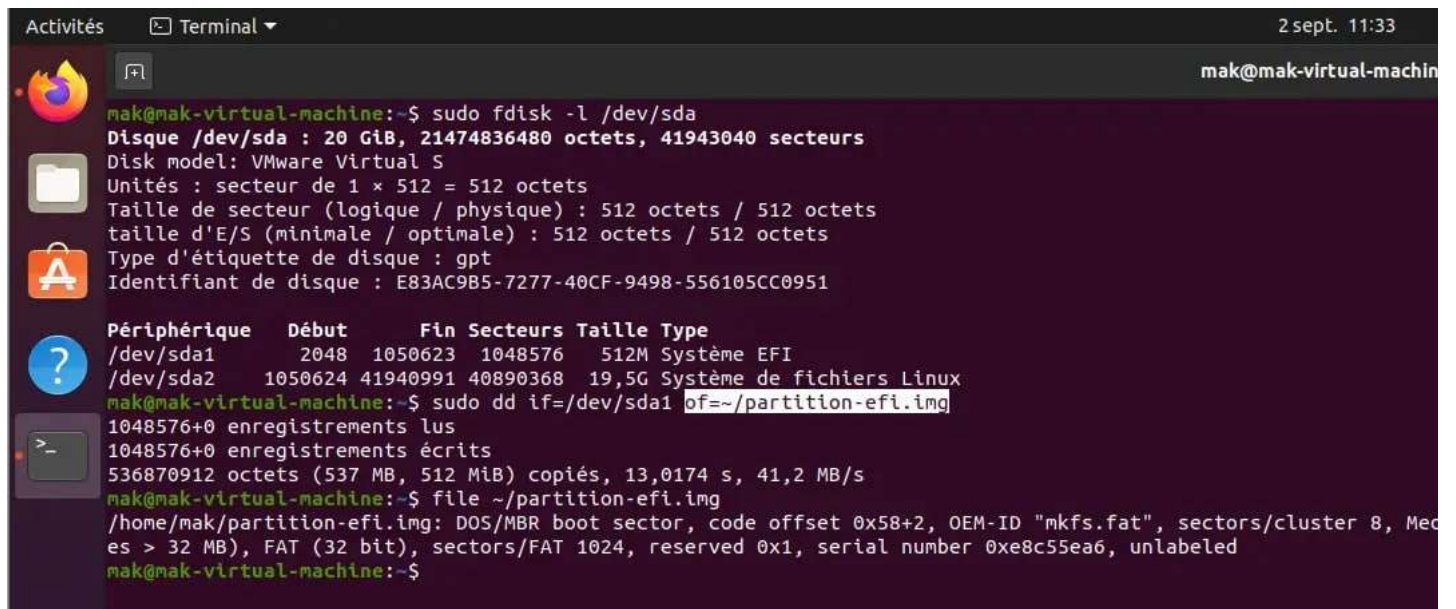
Bien sûr, on peut très bien ne sauvegarder qu'une partition de disque.
Il suffit de spécifier son numéro :

```
dd if=/dev/sda1 of=/partition.img
```

Enfin pour restaurer l'image et sauvegarde de disque à partir du fichier vers /dev/sdb :

```
dd if=/partition.img of=/dev/sdb
```

Par exemple, ci-dessous, on utilise dd pour **copier la partition EFI** vers un fichier. Cela permet d'effectuer une sauvegarde de la partition avant modification. On voit que le fichier .img est alors reconnu comme tel par la commande file.



```
mak@mak-virtual-machine:~$ sudo fdisk -l /dev/sda
Disque /dev/sda : 20 GiB, 21474836480 octets, 41943040 secteurs
Disk model: VMware Virtual S
Unités : secteur de 1 x 512 = 512 octets
Taille de secteur (logique / physique) : 512 octets / 512 octets
taille d'E/S (minimale / optimale) : 512 octets / 512 octets
Type d'étiquette de disque : gpt
Identifiant de disque : E83AC9B5-7277-40CF-9498-556105CC0951

Périphérique Début Fin Secteurs Taille Type
/dev/sda1 2048 1050623 1048576 512M Système EFI
/dev/sda2 1050624 41940991 40890368 19,5G Système de fichiers Linux
mak@mak-virtual-machine:~$ sudo dd if=/dev/sda1 of=~ /partition-efi.img
1048576+0 enregistrements lus
1048576+0 enregistrements écrits
536870912 octets (537 MB, 512 MiB) copiés, 13,0174 s, 41,2 MB/s
mak@mak-virtual-machine:~$ file ~/partition-efi.img
~/partition-efi.img: DOS/MBR boot sector, code offset 0x58+2, OEM-ID "mkfs.fat", sectors/cluster 8, Media > 32 MB), FAT (32 bit), sectors/FAT 1024, reserved 0x1, serial number 0xe8c55ea6, unlabeled
mak@mak-virtual-machine:~$
```

Là aussi, un article existe :

[Comment créer une image système avec Linux](#)

Créer un fichier ISO avec dd

De la même manière, on peut créer un fichier ISO d'un DVD-Rom, CD-Rom avec dd. C'est exactement le même principe avec en source le disque amovible et en destination le fichier iso.

```
dd if=/dev/cdrom of=~ / image - cdrom.iso bs=2048 conv=notrunc
```

Vider le contenu d'un disque

Vous souhaitez vendre votre PC et pour cela vider le contenu entier. Cela est aussi possible avec dd. Ainsi pour remplacer tous les secteurs d'un disque ou partition par des zéro :

```
dd if=/dev/zero of=/dev/sda
```

Ou encore, on peut aussi écrire des données aléatoires dans le disque :

```
dd if=/dev/urandom of=/dev/sda
```

Enfin on peut alors créer une boucle afin d'effectuer plusieurs passages pour s'assurer que la récupération de données est impossible. Par exemple pour 5 passages consécutifs :

```
for passage in `seq 5`; do dd if=/dev/urandom of=/dev/sda bs=8b conv=notrunc; done
```

Paramètres de la commande dd

Comme toutes commandes Linux, dd propose plusieurs options et paramètres. Pour afficher la liste des commandes :

```
dd --help
```

Les plus importantes :

- **bs**=OCTETS lire et écrire jusqu'à OCTETS octets à la fois (défaut: 512);
- **neutralise** ibs et obs
- **cbs**=OCTETS convertir OCTETS octets à la fois
- **conv**=CONVS convertir le fichier selon la liste des symboles séparés par des virgules
- **count**=N ne copier que N blocs d'entrée
- **ibs**=OCTETS lire jusqu'à OCTETS octets à la fois (512 par défaut)
- **if**=FICHIER lire FICHIER au lieu de l'entrée standard (stdin)
- **iflag**=ATTRS lire selon la liste de symboles séparés par des virgules
- **obs**=OCTETS écrire OCTETS octets à la fois (512 par défaut)
- **of**=FICHIER écrire dans FICHIER au lieu de la sortie standard (stdout)
- **oflag**=ATTRS écrire selon la liste de symboles séparés par des virgules
- **seek**=N se déplacer de N blocs de taille « obs » au début de la sortie
- **skip**=N ignorer N blocs de taille « ibs » au début de l'entrée
- **status**=QUELS les renseignements à afficher sur la sortie d'erreur standard :
 - « none » supprimer tout sauf les messages d'erreurs,
 - « noxfer » supprimer les statistiques de transfert final
 - « progress » montrer les statistiques de transfert périodique

conv

Le paramètre **conv** propose beaucoup d'options dont voici la liste :

- **ascii** de l'EBCDIC vers l'ASCII
- **ebcdic** de l'ASCII vers l'EBCDIC
- **ibm** de l'ASCII vers l'EBCDIC alternatif
- **block** remplir les enregistrements terminés par une nouvelle ligne par des espaces jusqu'à la taille « cbs »
- **unblock** remplacer les espaces en fin d'enregistrements de taille « cbs » par une nouvelle ligne
- **lcase** transformer les majuscules en minuscules
- **ucase** transformer les minuscules en majuscules
- **sparse** essayer de chercher plutôt qu'écrire la sortie pour les blocs
- d'entrée NUL
- **swab** inverser chaque paire d'octets en entrée
- **sync** remplir chaque bloc lu par des NUL jusqu'à la taille « ibs » ; avec
 - « block » ou « unblock », remplir avec des espaces au lieu de NULL
- **excl** échouer si le fichier de sortie existe déjà
- **nocreat** ne pas créer le fichier de sortie
- **notrunc** ne pas tronquer le fichier de sortie
- **noerror** continuer même après des erreurs de lecture
- **fdatasync** écrire physiquement les données en sortie avant la fin
- **fsync** identique, mais écrire aussi les métadonnées

dd prendre d'autres paramètres comme :

Pour copier, disque dur sur disque dur à l'aide de la commande dd donnée ci-dessous, l'option **sync** vous permet de tout copier à l'aide d'E / S synchronisées. Enfin **noerror** permet de continuer en cas d'erreur lors de la copie par dd.

```
dd if = /dev/sda of = /dev/sdb conv=noerror,sync
```

Enfin on peut utiliser **status=progress** pour afficher la progression avec de la vitesse de transfert et de copies.

bs – block size

bs (pour block size) indiquent à dd combien de secteurs doivent être copiés à la fois, que ce soit pour l'entrée, la sortie ou les deux. Par défaut, la plupart des versions de dd utilisent une taille de bloc de 512 octets à la fois pour l'entrée et la sortie.

Ce paramètre influe sur la vitesse de copie des données.

Avant 1999 lorsque la plupart des disques durs avaient une taille de secteur de 512 octets, mais ces dernières années, la plupart des disques durs ont une taille de secteur d'au moins 4 Ko (4 096 octets). Ce changement peut sembler insignifiant, mais peut conduire à d'énormes inefficacités lorsqu'il est combiné au fait que de nos jours, de nombreux disques durs grand public ont plus d'un téraoctet de capacité. Lorsque vous traitez avec un téraoctet ou plus de données, vous voulez vraiment vous assurer de choisir une taille de bloc optimale.

Quelle est la meilleure valeur de bs ?

Pour de meilleures performances, il faut régler la valeur bs sur la taille du cache disque.

Pour l'obtenir :

```
sudo hdparm -i /dev/sda
```

Mais on peut aussi utiliser [ce script](#) pour calculer la meilleure valeur de bs. Ce dernier affiche le débit pour différentes valeurs bs de dd. On multiplie facilement le débit par 1000 avec une bonne valeur de bs.

```
Activités Terminal 2 sept. 12:10 mak@mak-virtual-machine:
mak@mak-virtual-machine:~$ cd /tmp/
mak@mak-virtual-machine:/tmp$ wget https://raw.githubusercontent.com/tdg5/blog/master/_includes/scripts/dd_ibs_
--2020-09-02 12:10:01-- https://raw.githubusercontent.com/tdg5/blog/master/_includes/scripts/dd_ibs_test.sh
Résolution de raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.120.133
Connexion à raw.githubusercontent.com (raw.githubusercontent.com)|151.101.120.133|:443... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : 1553 (1,5K) [text/plain]
Enregistre : «dd_ibs_test.sh»

dd_ibs_test.sh 100%[=====]
2020-09-02 12:10:01 (3,88 MB/s) - «dd_ibs_test.sh» enregistré [1553/1553]

mak@mak-virtual-machine:/tmp$ chmod +x dd_ibs_test.sh
mak@mak-virtual-machine:/tmp$ ./dd_ibs_test.sh
NOTE: Kernel cache will not be cleared between tests without sudo. This will likely cause inaccurate results.
Generating test file...
block size : transfer rate
512 : 204 MB/s
1024 : 269 MB/s
2048 : 669 MB/s
4096 : 1 GB/s
8192 : 6 GB/s
16384 : 4 GB/s
32768 : 2 GB/s
65536 : 3 GB/s
131072 : 2 GB/s
262144 : 8 GB/s
524288 : 9 GB/s
1048576 : 9 GB/s
2097152 : 8 GB/s
4194304 : 0 GB/s
8388608 : 9 GB/s
16777216 : 9 GB/s
33554432 : 7 GB/s
67108864 : 359 MB/s
mak@mak-virtual-machine:/tmp$
```

Enfin on indique la taille du bloc dans dd de cette manière :

- bs=4k pour un bloc de 4 ko
- bs=1M pour un block de 1 Mo

Par exemple :

```
dd if = /dev/sda of = /dev/sdb conv=noerror,sync bs=4G
```

Liens

- [Comment créer une image système avec Linux](#)
- [GParted : redimensionner \(agrandir/réduire\) une partition de disque](#)
- [Comment retrouver/récupérer un fichier supprimé sur Linux](#)
- [Photorec : récupérer des fichiers supprimés sur Linux](#)
- [ddrescue : Comment récupérer les données d'un disque endommagé](#)
- [DiskGenius : vérifier et réparer les secteurs défectueux sur Windows 10](#)
- [chkdsk : réparer les erreurs de disque/lecteur sur Windows](#)
- [Formatage de disque, formatage rapide, formatage bas niveau](#)
- [BSOD : Vérifier les pilotes de Windows 10](#)
- [DiskPart : créer, supprimer des partitions de disque](#)

Tagged [linux](#), [cloner disque dur](#)

Navigation de l'article