

dd

Linux commande

dd permet de **copier** et **convertir des fichiers de type block**

dd signifie **Data Destroyer** ou **Data Dump** ou **Dataset Definition**

Il est possible d'utiliser la commande **ddrescue** qui reprend la copie en cas d'arrêt intempestif.

dd permet de reproduire des zones de disque qui ne font pas partie d'un système de fichier tels que le **secteur de démarrage** (le MBR), les **tables de partition**, les **traces laissées sur le disque par des fichiers effacés**, etc... L'un de ses emplois les plus importants est donc la création d'une copie de sauvegarde exacte d'une partition.

Le programme dd lance deux tâches asynchrones, d'une part de lecture et d'autre part d'écriture, contrairement à cp qui copie un fichier de façon séquentielle (lecture d'une partie du fichier sur un premier périphérique, puis copie sur le second).

1. Exemples d'utilisation

1.1 Créer une clé USB bootable

Pour créer une clé USB bootable à partir d'une image ISO.

On considère :

- /dev/sdX la clé non montée
- debian.iso est l'image ISO présent dans le dossier courant

```
# dd if=debian.iso of=/dev/sdX bs=1M; sync
```

1.2 Réaliser une image d'un DVD

Crée une image ISO de sauvegarde d'un CD ou d'un DVD :

On considère :

- /dev/cdrom le chemin d'accès au DVD
- debian.iso est le nom de l'image ISO

```
$ dd if=/dev/cdrom of=debian.iso
```

1.3 Découper un fichier

Pour découper un fichier en tranches de 100 Mio.

On considère :

- `grosfichier` le nom du fichier à découper
- `grosfichier-part1` le nom du 1er fichier de 100 Mio.
- `grosfichier-part2` le nom du 2nd fichier des 100 Mio suivant.

```
$ dd bs=10K count=10K if=grosfichier of=grosfichier-part1
$ dd bs=10K count=10K skip=10K if=grosfichier of=grosfichier-part2
$ dd bs=10K count=10K skip=20K if=grosfichier of=grosfichier-part3
$ dd bs=10K count=10K skip=30K if=grosfichier of=grosfichier-part4
```

Remarque, la commande `split` aurait été plus appropriée.

1.4 Copier la table MBR

Pour faire une copie du MBR (table des partitions principales comprises) d'un disque.

On considère :

- `/dev/hda` le disque dur contenant le MBR
- `full-mbr.bin` le nom du fichier contenant le MBR sauvegardé

```
$ dd if=/dev/hda of=full-mbr.bin bs=512 count=1
```

1.5 Effacer un disque dur avec des zéros

Pour détruire les données d'un disque dur `hda` en le remplissant de zéros.

```
$ dd if=/dev/zero of=/dev/hda
```

Remarque, la commande `shred` aurait été plus approprié.

1.6 Tester le débit des disques dur avec dd

Sous Linux, la commande `dd` peut être utilisée pour une mesure de performance en lecture et écriture séquentielle.

Pour se concentrer sur l'écriture des données sur un disque, la source des informations sera une suite de zéro disponible depuis le chemin `/dev/zero`. Si nous voulions mesurer la performance du disque, il aurait fallu écrire directement sur le chemin du disque (par exemple `of=/dev/sda`), mais cela effacerait le contenu du disque. En indiquant le chemin d'un fichier `of=/media/disk_6/test`, nous devons passer par le système de gestion de fichiers, qui peut nous ralentir. Mais cela n'enlèvera pas les conditions réelles que nous pouvons avoir avec un disque.

En utilisant `if=/dev/zero` et `bs=1G`, Linux aura besoin de 1GB d'espace disponible dans la RAM. Si vous n'avez pas suffisamment d'espace disponible, pensez à réduire cette valeur, par exemple à 512MB.

```
$ dd if=/dev/zero of=/media/disk_6/testfile bs=1G count=1 oflag=direct
```

2. Biblio

- <https://www.gnu.org/software/coreutils/dd>
- [https://fr.wikipedia.org/wiki/Dd_\(Unix\)](https://fr.wikipedia.org/wiki/Dd_(Unix))

From:

<https://www.abonnel.fr/> - **notes informatique & technologie**

Permanent link:

<https://www.abonnel.fr/informatique/linux/commandes/dd>

Last update: **2023/01/11 18:37**

