dd

- Objet : dd
- Niveau requis :
  débutant, avisé
- Commentaires : *Copier tout ou partie d'un disque par blocs d'octets.*
- Débutant, à savoir : Utiliser GNU/Linux en ligne de commande, tout commence là !
- Suivi :
  - Création par deuchdeb le 09-07-2010
  - Testé par deuchdeb le 02-09-2010
  - Complété par fiche le 15-08-2019
- Commentaires sur le forum : Lien vers le forum concernant ce tuto[1)]

ATTENTION ! La commande dd est très utile pour la sauvegarde de données, mais **elle peut avoir des effets désastreux si vous ne portez pas très attention à ce que vous faites**!

Introduction

La commande dd permet de copier tout ou partie d'un disque par blocs d'octets, indépendamment de la structure du contenu du disque en fichiers et en répertoires.

La copie avec dd permet de reproduire des zones de disque qui ne font pas partie d'un système de fichier : secteur de démarrage (le MBR), tables de partition, 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 de votre partition système et sa récupération*, par l'entremise d'un *live-CD* ou *live-DVD* en cas de pépin.

On peut utiliser dd pour bien d'autres choses : recherche dans les fichiers effacés, recherche dans la mémoire vive, création de disque virtuel, suppression des données d'un disque dur avant le don, revente d'un PC, etc...

Les bases

La structure de la commande a la forme générale suivante :

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

1. **if**: input file, partition à sauvegarder.
2. **of**: ouput file, destination de la partition sauvegardée.

source *représente les données à copier,* ***cible*** *est l'endroit où les copier.*

status-progress Affichage d'une commande en cours

Par défaut dd affiche déjà le temps d'exécution et la vitesse de transfert à la fin, mais le terminal peut afficher le déroulement du transfert en temps réel :

```
dd if=/dev/sda of=/dev/sdb conv=noerror status=progress
```

Source sur le forum : https://debian-facile.org/viewtopic.php?pid=325112#p325112

Merci **Antic Bomber** et **èfpé**.

Copier une partition de disque dur sur un autre disque dur

Remplacer *sda2, sdb2* par les valeurs qui correspondent à votre système : par exemple: *hdb1, hda1 ect...*

```
dd if=/dev/sda2 of=/dev/sdb2 bs=4096 conv=notrunc,noerror
```

sda2 et sdb2 sont des partitions. Vous voulez copier sda2 dans sdb2. Si sdb2 n'existe pas, dd commencera au début du disque et le créera.
bs *représente un nombre d'octets (par exemple: 512, 1024, 2048, 4096, 8192, 16384.)*

Cloner un disque dur en entier

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

- **notrunc** signifie « ne pas tronquer le fichier en sortie ».
- **noerror** signifie « de continuer en cas d'erreur ». Normalement, **dd** s'arrête en cas d'erreur. Si vous vous interrogez sur un disque dur, si vous vous demandez s'il fonctionne ou pas, vous pouvez tenter de l'utiliser, à titre de test, comme lecteur source de la commande dd. Vous devriez obtenir une erreur s'il ne fonctionne pas.
  Les lecteurs cibles en revanche doivent être vraiment en très mauvais état pour provoquer une erreur due à dd.

ATTENTION ! Il faut que les disques soient au moins de même volume pour le clonage.

IMPORTANT: Les disques ne doivent pas être montés!!Utiliser un système live pour travailler.

Créer un disque de sauvegarde

Copier son disque de travail en cours sda vers le disque sdb.

```
dd if=/dev/sda of=/dev/sdb bs=4096 conv=notrunc,noerror
```

Si je détruis l'installation sur sda, je n'ai qu'à booter avec un CD DebianLive et copier sdb vers sda :

```
dd if=/dev/sdb of=/dev/sda bs=4096 conv=notrunc,noerror
```

Je veux copier ma partion root dans mon home

```
dd if=/dev/sda2 (root) of=/home/user/root.img bs=4096 conv=notrunc,noerror
```

pour réaliser un backup de root faire :

```
dd if=/home/user/root.img of=/dev/sda2 (root) bs=4096 conv=notrunc,noerror
```

Remettre l'image de root dans la partition root si vous avez semé la pagaille et ne pouvez plus lancer le serveur X, ou éditer /etc/fstab, et ne trouvez pas ce que vous avez fait de mal. Cela ne prend que quelques minutes pour restaurer une partition root de 15 Go depuis un fichier image.

Créer une clé installation debian bootable

Voir le tuto :
Cle USB Installation debian

Copier une partition de disque dans un fichier placé sur une partition différente

Attention!! Ne pas copier une partition sur la même partition.

```
dd if=/dev/sdb2 of=/home/user/partition.image bs=4096 conv=notrunc,noerror
```

Cela créera un fichier qui est l'exacte réplique de la partition sdb2. Vous pouvez remplacer par hdb, sda, hda ou toute autre désignation de votre disque. Ou :

```
dd if=/dev/sdb2 ibs=4096 | gzip > partition.image.gz conv=noerror
```

Crée une archive gzip de la partition complète. Pour la restauration, utiliser :

```
dd if=/dev/sdb2 ibs=4096 | gunzip > partition.image.gz conv=noerror
```

Pour bzip2 (plus lent, plus petit), remplacer par bzip2 et bunzip2, et nommer le fichier

```
.bz2
```

Restaurer une partition de disque depuis un fichier image

```
dd if=/home/user/partition.image of=/dev/sdb2 bs=4096 conv=notrunc,noerror
```

De cette façon, vous pouvez avoir un grand disque dur et le partitionner, ainsi vous pouvez sauvegarder votre partition root. Si vous semez la pagaille dans votre partition root, vous n'avez qu'à booter depuis le CD DebianLive (ou un live CD quelconque) et restaurer l'image.

Disquettes

Copier une disquette sur le disque dur

```
dd if=/dev/fd0 of=/home/user/floppy.image bs=2x80x18b conv=notrunc
```

ou

```
dd if=/dev/fd0 of=/home/user/floppy.image conv=notrunc
```

18b représente la taille en octets de 18 secteurs de 512 octets chacun, 80x multiplie par le nombre de cylindres, 2x multiplie par le nombre de têtes - au total 1474560 octets - . Cela paramètre une unique requête de lecture de 1474560 octets sur /dev/fd0 et une unique requête d'écriture sur /home/user/floppy.image.

Il faut comprendre que, dans le cas d'une disquette, pour chacun des 80 cylindres supposés, 2 têtes (fictives) lisent chacune une piste de 18 secteurs... Pour y voir plus clair en ce qui concerne têtes, cylindres et secteurs voir plus haut.

Cela crée une image de la disquette sur le disque dur, laissant intact l'info de boot.

Le second exemple utilise le **bs=** par défaut de 512, qui est la taille du secteur d'une disquette.

Copier l'image d'une disquette du disque dur vers une disquette

Faire :

```
dd if=/home/user/floppy.image of=fd0 bs=2x80x18b conv=notrunc
```

Sauvegarde de disquettes sur disque dur

Je sauvegarde toutes mes disquettes sur disque dur. Les disquettes ne sont pas éternelles, aussi je fais :

```
dd if=/dev/fd0 of=/home/sam/disquettes/backup.bin conv=notrunc
```

Si ma disquette défaille, je peux faire des copies en nombre illimité :

```
dd if=/home/sam/disquettes/backup.bin of=/dev/fd0 conv=notrunc
```

Supprimer les données sur un disque dur

La commande dd permet également d'effacer les données présentes sur un disque avant de donner ou vendre son PC. Dans les commandes ci-dessous, sdX correspond au disque (en principe, le disque interne correspond à sda).

```
sudo dd if=/dev/zero of=/dev/sdX bs=4096 status=progress
```

Et pour que ce soit plus rapide :

```
sudo dd if=/dev/zero of=/dev/sdX bs=1M status=progress
```

Les Options

Option	Utilisation
if=fichier	Lire les données depuis le fichier indiqué plutôt que depuis l'entrée standard.
of=fichier	Écrire les données dans le fichier mentionné, et non pas sur la sortie standard. Si conv=notrunc n'est pas indiqué, le fichier est initialement tronqué à la taille spécifiée par seek= (0 octets si seek= n'est pas fourni).
ibs=nombre	Lire le nombre indiqué d'octets en une fois. Par défaut 512.
obs=nombre	Écrire le nombre indiqué d'octets en une fois. Par défaut 512.
bs=nombre	Lire et écrire le nombre indiqué d'octets en une fois. A priorité sur ibs et obs. (et indiquer bs n'est pas équivalent à indiquer la même valeur pour ibs et obs du moins lorsqu'aucune autre conversion que sync, noerror et notrunc n'est indiquée, car cela indique que chaque bloc d'entrée doit être copié dans un bloc de sortie indépendant, sans regrouper les blocs plus courts).
cbs=nombre	Indique la taille des blocs pour les conversion block et unblock.
skip=nombre	Ignorer le nombre indiqué de blocs (dont la taille est fournie par ibs) au début de la lecture.
seek=nombre	Ignorer le nombre indiqué de blocs (dont la taille est fournie par ibs) au début de l'écriture.
count=nombre	Copier seulement le nombre indiqué de blocs (dont la taille est fournie par ibs), et non pas tout jusqu'à la fin du fichier.
conv=conversion[,conversion]...	Modifier le fichier comme indiqué par l'argument conversion, qui peut prendre les valeurs suivantes (pas d'espace autour des virgules lorsque plusieurs arguments sont fournis) ainsi : **ascii** Convertir l'EBCDIC en ASCII. **ebcdic** Convertir l'ASCII en EBCDIC. **ibm** Convertir l'ASCII en EBCDIC IBM.
block	Compléter les blocs se terminant par un saut de ligne avec des espaces, jusqu'à atteindre la taille mentionnée par cbs.

Option	Utilisation
unblock	Remplacer les espaces en fin de blocs (de taille cbs) par un saut de ligne.
lcase	Transformer les majuscules en minuscules.
ucase	Transformer les minuscules en majuscules.
swab	Échanger par paire les octets lus en entrée. Contrairement à la commande dd d'Unix, la version GNU fonctionne également lorsqu'on copie un nombre impair d'octets. Dans ce cas, le dernier octet est tout simplement copié. [POSIX 1003.2b, interprétations PASC 1003.2 #3 and #4]
noerror	Continuer même après des erreurs de lecture.
notrunc	Ne pas limiter la taille du fichier de sortie.
sync	Compléter chaque bloc lu avec des NULs pour atteindre la taille ibs.

Les variables d'environnement

Les variables d'environnement suivantes affectent les messages et les messages d'erreur de dd :

LANG
Fournit une valeur par défaut pour les variables d'internationalisation qui sont non définies ou nulles. Si LANG est non défini ou nul, la valeur correspondante par défaut dépendante de l'implantation locale sera utilisée. Si chacune des variables d'internalisation contient une valeur non valide, l'utilitaire se comportera comme si aucune des variables n'avait été définie.

LC_ALL
Si positionnée sur une valeur de chaine de caractères non vide, prend le pas sur les valeurs de toutes les autres variables d'internationalisation.

LC_CTYPE
Définit la locale utilisée pour l'interprétation comme caractères des séquences d'octets de données texte (par exemple, caractères à octet unique, par opposition aux caractères multi-octets dans les arguments ou les fichiers d'entrée), la classification des caractères en majuscules ou minuscules, et le mappage des caractères d'une casse à une autre.

LC_MESSAGES
Détermine la locale à utiliser pour déterminer le format et le contenu des messages de diagnostic écrits dans la sortie standard des erreurs et les messages informatifs écrits dans la sortie standard.

NLSPATH
Détermine l'endroit des catalogues de messages pour le traitement de LC_MESSAGES.

Récupérer les données d'un disque

- ddrescue

Lien sur l'utilisation de dd sur le forum

- https://debian-facile.org/viewtopic.php?pid=200330#p200330
- https://debian-facile.org/viewtopic.php?pid=255790#p255790 Clé bootable pour installation.

Liens externes

- Man de dd en français
- La source de ce document, trés complet

[1)]

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

From:
http://debian-facile.org/ - **Documentation - Wiki**

Permanent link:
http://debian-facile.org/doc:systeme:dd

Last update: **25/02/2020 09:13**