Logiciel

dd : copier et convertir un disque ou fichier

La commande **dd** permet de copier des fichiers ou disque entier par blocs et même les convertir.

Cet outil très puissant doit être utilisé avec précaution.

Idéal pour sauvegarder le “mbr”, créer des “iso” …

Introduction

Pré-requis

Installation

Cette commande fait partie d'Ubuntu. Aucune installation nécessaire

Configuration

Utilisation

Syntaxe de base

```
$ sudo dd if=[fichier_source] of=[fichier_cible] [Options]
```

if=fichier_source
: abréviation de **Input File**, source : un fichier, un répertoire, un blocs du disque, un périphérique…

of=fichier_cible
: abréviation de **Output File**, emplacement de sortie, où l'on va écrire.

Options

bs=taille_des_blocs
: abréviation de **Block Size**, taille des blocs pour copier les données, par défaut en octets, pour Ko ajouter un “K”, pour Mo ajouter un “M”…

count=nombre_de_blocs
: nombre de blocs à copier, vous pouvez ajouter un “K” pour “1 000”, un “M” ajouter un “1 000 000”…

skip=nombre_de_blocs_a_sauter
: nombre de blocs à sauter au début du fichier.

ibs=taille_des_blocs_a_l'entrée
: taille des blocs en entrée en octets. Par défaut 512 octets. Nécessite **obs**.

obs=taille_des_blocs_en sortie
: taille des blocs en sortie en octets. Par défaut 512 octets. Nécessite **ibs**.

seek=nombre_de_blocs_a_sauter
: nombre de blocs à sauter au début du fichier dont la taille est fournie par **ibs**.

conv=type_1,type_2
: Permet de faire une ou plusieurs conversions d'un type vers un autre.
Liste des types de conversion:

- ascii
  ⇒ EBCDIC vers ASCII
- ebcdic
  ⇒ ASCII vers EBCDIC
- ibm
  ⇒ EBCDIC vers ASCII
- lcase
  ⇒ Convertit en minuscules
- noerror
  ⇒ continuer même s'il y a des erreurs de lecture
- ucase
  ⇒ Convertit en majuscules
- swab
  ⇒ Échanger par paire les octets lus en entrée
- sync
  ⇒ Compléter chaque blocs d'entrée avec des zéros pour avoir la bonne taille de blocs en “ibs”

Exemples

- sauvegarder les 512 premiers octets (bs=512) du disque **sda1**, en une seule partie (count=1), dans le fichier **boot.mbr** :

```
$ sudo dd if=/dev/sda1 of=boot.mbr bs=512 count=1
```

- convertir en **ascii** le fichier **/home/totoc** en **/home/totox** :

```
$ sudo dd if=/home/totoc of=/home/totox conv=ascii
```

- découper le fichier **fichier_a_découper** en morceaux de 10Mo ; à chaque ligne, on fait 1000 blocs(count=1K) de 10Ko(bs=10K), pour sauvegarder en le fichier **fichier_a_découper_XXXème_partie**, à partir de la 2ème ligne on saute les parties déjà découpées avec skip=1K :

```
$ sudo dd bs=10K count=1K if=fichier_a_découper
of=fichier_a_découper_1ère_partie
```

```
$ sudo dd bs=10K count=1K skip=1K if=fichier_a_découper
of=fichier_a_découper_2ème_partie
$ sudo dd bs=10K count=1K skip=2K if=fichier_a_découper
of=fichier_a_découper_3ème_partie
$ sudo dd bs=10K count=1K skip=3K if=fichier_a_découper
of=fichier_a_découper_4ème_partie
$ sudo dd bs=10K count=1K skip=4K if=fichier_a_découper
of=fichier_a_découper_5ème_partie
```

Désinstallation

Voir aussi

- **(fr)** http://www.octetmalin.net/linux/tutoriels/dd-copier-et-convertir-disque-ou-fichier-par-blocs-blocks.php

---

Basé sur « Article » par Auteur.

From:
https://nfrappe.fr/doc-0/ - **Documentation du Dr Nicolas Frappé**

Permanent link:
https://nfrappe.fr/doc-0/doku.php?id=logiciel:systeme:dd:start

Last update: **2022/08/13 21:57**