HOWTO zum Kommando "dd" (disk duplicate)

(C) 2006-2023 T.Birnthaler/H.Gottschalk <howtos(at)ostc.de>
OSTC Open Source Training and Consulting GmbH
http://www.ostc.de

$Id: dd-HOWTO.txt,v 1.12 2025/02/18 10:07:50 tsbirn Exp $

Dieses Dokument beschreibt den Datenbackup und -restore mittels "dd".

INHALTSVERZEICHNIS

0) Name
1) Beschreibung
2) Beispiele

---

0) Name

Über die Bedeutung des Namens "dd" streiten sich die Geister. Angeblich war zuerst "cc" = "copy and convert" angedacht, leider hieß der C-Compiler bereits so. "dd" steht (etwas umständlich und mißverständlich) für "disk duplicate" oder für "dataset definition" (vom gleichnamigen IBM/360 JCL-Kommando "dd" mit ähnlichen Aufgaben, daher auch die ungewöhnliche Syntax ohne "-" als Optionskennzeichen).

1) Beschreibung

"dd" kopiert Daten "roh" (d.h. direkt sektor- oder recordweise ohne Verwendung eines Dateisystems) von/auf Speichermedien wie z.B. Festplattenpartitionen, CD-ROMs, Disketten und USB-Sticks. Es kann dabei:

  * Zeichen umkodieren (ASCII <-> EBCDIC)
  * Records umsetzen
    + Verlängern (auffüllen mit Leerzeichen/Null-Bytes)
    + Verkürzen (wegstreichen von Leerzeichen/Null-Bytes)
    + Umwandlung der Groß/Kleinschreibung
    + Vertauschen von Bytes

"dd" wird oft auch zum "Blocken" von Daten verwendet (vor allem bei Magnetbändern), um häufige Schreiboperationen kleiner Datenmengen zu seltenen Schreiboperationen großer Datenmengen zusammenzufassen. Die allgemeine Syntax von "dd" lautet (ACHTUNG: Optionen bei "dd" ohne Bindestrich angeben!):

```
dd [OPTION=VALUE ...]
```

Ein typischer Aufruf lautet z.B.:

```
dd if=INFILE of=OUTFILE \
   count=NOOFBLOCKS bs=BLOCKSIZE \
   skip=SKIPBLOCKS seek=SEEKBLOCKS
```

Ohne Angabe von Optionen kopiert "dd" beliebig viele Records von der Standard-Eingabe auf die Standard-Ausgabe (bis die Standard-Eingabe vollständig gelesen ist).

Mit Angabe von Optionen kopiert "dd" NOOFBLOCKS Datenblöcke (Sektoren) der Größe BLOCKSIZE (Std: 512) von INFILEPATH nach OUTFILEPATH (können auch Geräte sein!). Es überspringt dabei optional SKIPBLOCKS Blöcke des Eingabegerätes und SEEKBLOCKS Blöcke des Ausgabegerätes. Als Optionen sind möglich:

```
+-----------------+-----------------------------------------------------+
```

Option	Bedeutung
count=NOOFBLOCKS	Anzahl an zu kopierenden Blöcken (Std: Alle)
bs=BLOCKSIZE	Blockgröße (Eingabe + Ausgabe gemeinsam) (Std: 512)
if=INFILEPATH	Eingabe-Datei (Default: Standard-Eingabe)
of=OUTFILEPATH	Ausgabe-Datei (Default: Standard-Ausgabe)
ibs=BLOCKSIZE	Eingabe-Blockgröße (Std: bs)
obs=BLOCKSIZE	Ausgabe-Blockgröße (Std: bs)
cbs=BLOCKSIZE	Konvertierungs-Blockgröße (Std: bs)
conv=KEYWORDS	Konvertierungen (durch Komma getrennt: ascii, ebcdic, ibm, block, unblock, lcase, ucase, swab, nocreat, notrunc, noerror, sync, fdatasync, fsync)
skip=SKIPBLOCKS	Blöcke (bs/ibs) des Eingabe-Gerätes überspringen
seek=SEEKBLOCKS	Blöcke (bs/obs) des Ausgabe-Gerätes überspringen
status=noxfer	Übertragungsstatistik am Ende nicht ausgeben
iflag=FLAGS	Eingabe-Flags (durch Komma getrennt: append, direct, dsync, sync, nonblock, nofollow, noctty)
oflag=FLAGS	Ausgabe-Flags (kommaseparierte Liste)

ACHTUNG: Durch den direkten Gerätezugriff ist bei Tippfehlern sehr schnell der Inhalt eines Boot-Records oder einer Partition/Festplatte zerstört (z.B. count vergessen, if/of vertauscht, ...). Daher am besten die Optionen "if" und "of" erst am Ende der Optionsliste angeben.

ACHTUNG: Eine BLOCKSIZE von 1 führt zu einem sehr langsamen Ablauf von "dd". Bei einigen Geräten (z.B. Magnetbändern) sind große Blockgrößen (z.b. bs=10k) nötig, um eine hohe Übertragungsgeschwindigkeit zu erreichen.

Folgende Einheiten bei den Blockgrößen sind in Form eines Suffixes möglich:

Einheit	Bedeutung
leer	Byte
c	Byte (character)
w	2 Byte (word)
b	512 Byte (block)
kB K	1000 bzw. 1024 Byte (Kilo-byte)
MB M	1000^2 bzw. 1024^2 Byte (Mega-byte)
GB G	1000^3 bzw. 1024^3 Byte (Giga-byte)
TB T	1000^4 bzw. 1024^4 Byte (Tera-byte)
PB P	1000^5 bzw. 1024^5 Byte (Peta-byte)
EB E	1000^6 bzw. 1024^6 Byte (Exa-byte)
ZB Z	1000^7 bzw. 1024^7 Byte (Zetta-byte)
YB Y	1000^8 bzw. 1024^8 Byte (Yotta-byte)

Zum Abschluss eines Kopiervorgangs gibt "dd" eine Ein/Ausgabstatistik auf STDERR aus:

```
1833+0 records in 1833+0 records  out  938767  bytes
(9.4 GB) copied, 34.6279 seconds, 271 MB/s
```

Der Wert 1833+0 bedeutet, dass 1833 vollständige und 0 teilweise Blöcke gelesen bzw. geschrieben wurden (tritt auf, wenn die Größe der kopierten Daten kein ganzahliges Vielfaches der Blockgröße ist)

Fortschrittsanzeige: Schickt man das Signal USR1 an einen dd-Prozess, so gibt er die aktuelle Ein/Ausgabestatistik auf STDERR aus:

```
  dd if=/dev/zero of=/dev/null &
  kill -USR1 $!                      # $! = PID des letzten Hintergrundprozesses


2) Beispiele
------------
* Master Boot Record (MBR) der 1. IDE-Festplatte auf Diskette sichern:
    dd count=1 bs=512 if=/dev/hda of=/media/floppy/mbr.sav

* Boot Record (BR) der 3. Partition der 2. IDE-Festplatte auf Diskette sichern:
    dd count=1 bs=512 if=/dev/hdb3 of=/media/floppy/b3.br

* Gesicherten MBR von Diskette zurückspielen:
    dd count=1 bs=512 if=/media/floppy/mbr.sav of=/dev/hda

* Gesicherten BR von Diskette auf 3. Partition der 2. IDE-Festpl. zurückspielen:
    dd count=1 bs=512 if=/media/floppy/b3.br of=/dev/hdb3

* 1. IDE-Partition (z.B. DOS/WIN95/WIN98/WINNT) komplett auf Datei (in anderer
  Partition!) sichern:
    dd if=/dev/hda1 of=/tmp/hda1.img            # unkomprimiert
    dd if=/dev/hda1 | gzip > /tmp/hda1.img      # komprimiert

* 1. IDE-Partition (z.B. DOS/WIN95/WIN98/WINNT) wiederherstellen:
    dd if=/tmp/hda1.img of=/dev/hda1            # unkomprimiert
    gunzip -c /tmp/hda1.img > /dev/hda1         # komprimiert (-c=console)

* Bootdiskette erzeugen (Kern-Image roh auf Diskette schreiben)
    dd bs=8192 if=/usr/src/linux/arch/i386/boot/bzImage of=/dev/fd0

* MBR roh auf Diskette sichern (Festplatte kann davon gebootet werden):
    dd count=1 bs=512 if=/dev/hda of=/dev/fd0

* Diskette 1:1 kopieren (per Disketten-Image)
  - 1. Diskette einlegen
      dd if=/dev/fd0 of=/tmp/floppy.image  # Disketten-Image nach /tmp kopieren
  - 2. Diskette einlegen
      fdformat /dev/fd0u1440               # Eventuell formatieren
      dd if=/tmp/floppy.image of=/dev/fd0  # Disketten-Image von /tmp auf Floppy
      rm /tmp/floppy.image                 # Eventuell Disketten-Image löschen

* CDROM/DVD-Image auf Festplatte kopieren (Laufwerk ist Master am 2. IDE-Kanal):
  - 1. CDROM/DVD reinstecken (NICHT montieren!)
      dd if=/dev/hdc   of=/tmp/cdrom.image    # Hardwarename (IDE)
      dd if=/dev/sr0   of=/tmp/cdrom.image    # Hardwarename (SATA)
      dd if=/dev/scd0  of=/tmp/cdrom.image    # Allgemeiner Name (symb. Link)
      dd if=/dev/cdrom of=/tmp/cdrom.image    # Allgemeiner Name (symb. Link)
      dd if=/dev/cdrw  of=/tmp/cdrom.image    # Allgemeiner Name (symb. Link)
      dd if=/dev/dvd   of=/tmp/cdrom.image    # Allgemeiner Name (symb. Link)
      dd if=/dev/dvdrw of=/tmp/cdrom.image    # Allgemeiner Name (symb. Link)

* Swap-Datei der Größe 10 MByte anlegen und verwenden:
    dd if=/dev/zero of=/tmp/swapfile count=10 bs=1M
    mkswap -c /tmp/swapfile 1000
    swapon /tmp/swapfile
```