dd (disk dump)

Linux Programm zur Sicherung von Festplatten und Partitionen

Inhalt:

1. Allgemeines

dd (Disk Dump) ist ein Programm zur Datensicherung.

Datensicherung ist **immer** ratsam.

Wer keine Datensicherung hatte, wird dies vielleicht schon schmerzlich erfahren haben, wenn seine wichtigen Dateien (Bilder, Videos, mp3´s, Texte, Kontoauszüge, E-Mails und Browserlinks) unwiederbringlich verloren waren.

Oder aber man muss sein Betriebssystem welches lange auf der Festplatte reifte neu installieren.

Hierzu kann das kostenlose und zu jeder Linuxdistribution (auch Live-CD) gehörende Programm dd benutzt werden.

Alle Angaben sind ohne Gewähr und sollten nur unter **genauester** Beachtung der Syntax von dd angewendet werden.

2. Funktionsweise

dd ist ein Programm zur Sicherung von Partitionen und ganzen Festplatten (Klonen; Duplizieren).

Hierbei wird eine Quelle (Partition oder ganzes Laufwerk (Festplatte, CD, DVD) und ein Ziel angegeben. Desweiteren kann die zu übertragende Datensatzgröße (Blockgröße) festgelegt werden, was sich bei der Übertragung mittels DFÜ als sehr nützlich erweisen kann.

Hinweis:

Da dd immer eine 1:1 Kopie der Quelle anlegt, benötigt das Ziellaufwerk soviel freien Speicherplatz wie die Quelle maximal aufweist.

Durch das Pipelining (weiterleitung eines Programmergebnisses an ein anderes Programm zur Weiterverarbeitung) können die Daten durch Nachschalten der gzip-Komprimierung Speicherplatzschonend gepackt werden.

Das Ergebnis wird in einer Datei abgelegt deren Name frei vergeben werden kann.

3. Installation

dd (Disk Dump) ist Bestandteil der Grundinstallation fast aller Distributionen, so auch Ubuntu. Somit ist eine nachträgliche Installation in der Regel nicht nötig.

4. Syntax

Die Grundsyntax von dd ist sehr einfach:

```
dd if of bs
```

Erklärung:

if	Inputfile (Eingabedatei/-Laufwerk)
of	Outputfile (Ausgabedatei/-Laufwerk)
bs	Blocksize/-größe (optional und erst mal nicht wichtig)

- Auf einer deutschen Tastatur erhält man das Zeichen "|" durch gleichzeitiges Drücken der AltGR (rechts neben der Leertaste) und der Taste für <> (links neben "Y").
- Da unter Linux jedes Device (Laufwerk etc.) eine Datei ist und auch so behandelt wird, wurde hierbei der Name "file" verwendet.

Das Eingabe "file" kann in der Regel eine Datei, Partition oder ein komplettes Laufwerk (Festplatte CD, DVD) sein.

5. Duplizieren einer Festplatte auf eine andere (Klonen)

Dies ist unabhängig vom Dateisystem, also geeignet für Windows, Linux und andere Dateisysteme.

Steht beispielsweise der Kauf einer neuen und größeren Festplatte an, so erledigt dd für uns das Klonen der kompletten Festplatte und erstellt eine lauffähige 1:1 Kopie der Alten auf die neue Festplatte.

Hierbei ist, z.b. die alte Festplatte an */dev/hda* und die Neue an */dev/hdb* angeschlossen (die Zielfestplatte muss mindestens über so viel Speicherplatz verfügen wie die alte Festplatte).

Damit das Ganze auch funktioniert, muss man die alte Festplatte entweder auf read only (nur lesen) setzen, oder besser:

Man benutzt eine LiveCD, z.B. von Ubuntu oder Knoppix.

Da man nur eine Kommandozeile (Terminal) benötigt, kann man auch ein Minimal-Linux vom USB Stick booten.

Die Zielfestplatte muss nicht formatiert werden, da die Inhalte 1:1 übertragen werden.

```
dd if=/dev/hda of=/dev/hdb
```

Die Angabe einer Blockgröße (bs) ist nicht notwendig.

Das Kopieren wird seine Zeit dauern. *Anschließend* jumpert man die zweite Festplatte um (von Slave nach Master) und baut diese anstelle der alten Festplatte wieder ein.

Es steht dann einem Boot nichts mehr im Wege und die neue Festplatte müsste wie die Alte funktionieren.

Ist die neue Festplatte grösser als die alte Festplatte, so verbleibt der freie Speicher zur weiteren Verfügung.

Man kann dann mittels gparted die Partitionsgrössen ggf. anpassen.

6. Eine Partition kopieren

Eine Partition kann wie eine komplette Festplatte gesichert werden, der Unterschied besteht lediglich darin, dass /dev/hda**1** als Inputfile und /dev/hdb**1** als Outputfile verwendet werden.

Zum Beispiel:

```
dd if=/dev/hda1 of=/dev/hdb1
```

Auch hier ist zu beachten, dass bei der Sicherung der Rootpartition diese nur read only gemountet ist oder eine Live CD verwendet wird!

7. Unkomprimierte Sicherung einer Partition

Diese Syntax beschreibt die Datensicherung einer Partition in eine Datei:

```
dd if=/dev/hda1 of=/dev/hda2/sicherung_hda1.file
```

Unkomprimierte Rücksicherung

Die Rücksicherung kann hier aus einer Datei gemacht werden. Hat man wie oben beschrieben eine komplette Partition als Partition gesichert, ist der Befehl von oben selbstverständlich nur umgekehrt auszuführen. Für Datei -> Partition gilt:

```
dd if=/dev/hda2/sicherung_hda1 of=/dev/hda1
```

8. Komprimierte Sicherung einer Partition

Diesmal wird das Inputfile (/dev/hda1 in dieseM Fall) als komprimierte Datei gesichert:

```
dd if=/dev/hda1 | gzip > /dev/hda2/sicherung_hda1.gz
```

Komprimierte Rücksicherung

Hierbei wird die gepackte Sicherung der Partition hda1 nach /dev/hda1 zurück geschrieben, nachdem es entpackt wurde:

```
gunzip -c /dev/hda2/sicherung_hda1.gz| dd of=/dev/hda1
```

9. Ganze Festplatte sichern

Eine Vollsicherung (Diskimage) sollte von Zeit zu Zeit erfolgen, damit man im Fehlerfall schnell und ohne zeitaufwendige Neuinstallation wieder eine lauffähige Version erhält.

Der Sicherungsrythmus hängt ganz von den individuellen Bedürfnissen ab (z.B. monatlich oder wöchentlich).

Möchte man eine **tägliche** Datensicherung der geänderten Anwenderdaten, empfiehlt sich dies z.B. mittels des Rsync.

Da es wenig Sinn macht, ein Diskimage unkomprimiert abzulegen, wird hier nur die komprimierte Form beschrieben. Unkomprimiert kann es wie o.a. durchgeführt werden.

Voraussetzung:
Die zu sichernde Festplatte muss entweder auf read-only (nur lesen) gestellt werden, besser aber benutzt man eine LiveCD wie die von Ubuntu oder Knoppix.

Da man nur eine Kommandozeile (Terminal) benötigt, kann man auch ein MinimalLinux vom USB-Stick booten.

Das Ziel sollte hierbei gemountet sein, über genügend Speicherplatz verfügen und auch die Dateigrösse an einem Stück aufnehmen können (man beachte die Einschränkung bei vfat (FAT32) auf 4 GByte pro Datei).

Diskimage der ersten Festplatte herstellen

Hier wird die erste Festplattre (hda) in komprimierter Form als Datei auf Partition hdb1 auf der zweiten Festplatte gesichert:

```
dd if=/dev/hda | gzip > /dev/hdb1/sicherung_hda.gz
```

Diskimage wieder zurückspielen

Hierbei verwendet man wie o.a. entweder eine LiveCD oder ein Minimal-Linux. Das Ziellaufwerk wird hierbei komplett überschrieben.

```
gunzip -c /dev/hdb1/sicherung_hda.gz| dd of=/dev/hda
```

10. Ganze Festplatte löschen

Der dd Befehl kann auch dazu benutzt werden, den gesamten Festplatteninhalt oder alle Daten auf einer Zielangabe zu löschen bzw. zu überschreiben.

Achtung!: Die vorhandenen Daten auf der Zielangabe werden **unwiederbringlich** gelöscht.

Hierbei wird das gesamte Laufwerk /dev/hda mit Zufallszahlen überschrieben. Je nach Grösse der Festplatte kann dies eine ganze Zeit dauern.

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

Führt man diesen Befehl ein paar mal hintereinander durch, ist es praktisch unmöglich, den ursprünglichen Inhalt der Daten wiederherzustellen.

11. Master Boot Reccord (MBR) sichern

Mittels nachfolgender Befehle kann der Bootsektor des MBR (Master Boot Record) gesichert und wieder zurückgespielt werden.

MBR sichern

Dafür muss man natürlich wissen, wo sich der MBR befindet. In der Regel wird dieser auf die erste Festplatte geschrieben, wo der Befehl dann lautet:

```
dd if=/dev/hda bs=512 count=1 of=/Ziel_und_Dateiname_bestimmen
```

MBR zurückspielen

Hierbei wird der MBR (wieder zurück) auf hda geschrieben:

```
dd if=Ziel_und_Dateiname_angeben of=/dev/hda
```

MBR löschen

Möchte man den kompletten MBR aus irgendeinem Grund löschen, macht man das hiermit:

```
dd if=/dev/zero of=/dev/hda1 bs=1 count=446 conv=notrunc
```

12. Weitere Möglichkeiten von dd

Wie man sieht, ist dd ein mächtiges Werkzeug und erspart einem den Kauf von Zusatzsoftware, wie dies bei anderen Betriebssystemen der Fall wäre.

Es besteht auch die Möglichkeit, dd über eine ssh-Verbindung zu tunneln und somit ein Backup über das Netzwerk oder das Internet durchzuführen.

Alle möglichen Befehle von dd können der **manpage** (man dd) entnommen werden, es wurden hier nur die am häufigsten benötigten Funktionen beschrieben.

Es ist anzuraten solch eine Rettungs-CD vor einem Crash schon mal vorsorgehalber zu erstellen und diese auch mal auszuprobieren.

Letzte Änderung 12.09.2009