



SV3: Linux Server

Komprimierungs- und Sicherungstools

Agenda

Komprimierungs- und Sicherungstools

- Komprimierungsarten
- Tar, zip, Gzip, bzip, dd
- Spiegeln von Festplatten mit dd
- Images mit dd erzeugen
- Aufbewahrungsorte für Sicherungen

Dateien in Archive zu verpacken und ihre Größe durch Kompressionsalgorithmen zu verkleinern ist ein häufiger Vorgang in der Datenverarbeitung.

Grundsätzlich unterscheidet man zwischen Archiven und komprimierten Dateien:

- Ein Archiv ist ein unkomprimierter Behälter für Dateien. Dieses Archiv kann durch Kompressionsverfahren komprimiert werden.
- Es gibt allerdings ein paar Formate, die gleich auch eine Komprimierung beinhalten.

Bildlich gesprochen funktioniert eine Datenkompression ungefähr so, wie wenn man ein Kopfkissen zusammendrückt.

- Die ganze Luft zwischen Kissenbezug und Federn wird nicht unbedingt benötigt.
- Durch das Zusammendrücken komprimiert man das Kissen, indem man die Luft entfernt (so gut es geht) – das Kissen wird kleiner, sprich komprimierter, als vorher.

So ähnlich verhält es mit der Datenkompression. Auch hier werden nicht unbedingt notwendige Elemente, in diesem Fall Daten, entfernt und der Speicherplatz, den die übrigen Daten noch benötigen, wird kleiner.

Verschiedene Kodierungsmethoden der Kompression

■ Verlustfreie Kompression

findet meist nur bei Textdateien statt.

Wenn Text als Zeichenkette und nicht als Bild oder ähnliches gespeichert wird, kann beispielsweise mithilfe der Wörterbuch-Methode Speicherplatz eingespart werden.

■ Verlustbehaftete Kompression (das MP3-Format)

Diese Vorgehensweise sorgt natürlich unter Umständen für enorm hohe Kompressionsraten.

Allerdings werden jene irrelevanten Informationen unwiderruflich gelöscht.

Somit ist nach einer Kompression der Ursprungszustand nicht wiederherstellbar.

Komprimierungstools

Hier werden wiederkehrende Zeichenkettenteile einfach durch Abkürzungen ersetzt.

Text:	Hallo Welt. Hallo User.
Kodierung:	x5 Welt. X5 User.

Mithilfe des Run Length Encoding (RLE) werden identische Textbestandteile, die hintereinander stehen, nur einmal abgespeichert. Auch so kann Speicherplatz gespart werden.

Text:	Heute war es schön, schön, schön, und morgen wird es schöner.
Kodierung:	Heute war es /3/schön,/ und morgen wird es schöner.

Komprimierungstools

Bei einer Entropiekodierung, wie es beispielsweise beim **Huffman-Code** oder der Arithmetischen Kodierung der Fall ist, werden Textelemente nach ihrer Häufigkeit binär sortiert und codiert, wobei das Element, das am häufigsten vorkommt, die kleinste binäre Zahl erhält.

Mittels jener Zahlen werden die Textelemente dann in einem eigenen **Wörterbuch** gespeichert.

Text:	WENN HINTER FLIEGEN FLIEGEN, FLIEGEN FLIEGEN FLIEGEN NACH.
Kodierung:	10 100 1 1 1 1 1 11

Unterschied zwischen verlustfreier und verlustbehafteter Kompression

- Der Vorteil der verlustfreien Kodierung ist klar:
Sie gewährleistet, dass man sich alle „wegkomprimierten“ Daten aus dem komprimierten Datenpaket erschließen und wiederherstellen kann.
- Neben der verlustfreien Kompression gibt es noch die verlustbehaftete Kompression.
Hier werden irrelevante Informationen nicht kodiert, sondern direkt entfernt. Daher spricht man auch von einer Irrelevanz Reduktion.

Was und worauf kann gesichert werden?

Wir können Sicherungen gliedern je nachdem, was und wann gesichert werden soll, zu sichern sind z.B.:

- Der aktuelle Stand während der Erstellung von Dateien und die vorhergehende Version einer geänderten Datei
- Bestimmte Dateien, z.B. Testprogramme, Texte zu einem Buch usw.
- Wöchentlich alle Dateien eines Benutzers oder einer Benutzergruppe oder aller Benutzer (Dateibaum) und täglich davon nur die veränderten Dateien
- Alle Benutzerprogramme (Anwenderprogramme), die meist ebenfalls unter bestimmten Directories abgelegt sind
- Systemsoftware , gesamte Platte(n), Betriebssystem, Rechner , usw.

Was und worauf kann gesichert werden?

In einer Umfrage zu »**Auf was können Sie sichern?**« wurden folgende Antworten gesammelt:

- auf Papier
- unter anderem Dateinamen
- in ein anderes Directory
- auf einen anderen Plattenbereich
- auf einen anderen Rechner (File-Server),
- am besten an einem anderen Ort – am einfachsten über ein Netzwerk (Cloud)
- auf USB-Stick oder entfernbare USB-Platte
- auf CD oder DVD brennen
- auf Magnetbändern

Einbinden von Wechselmedien

Die verschiedenen Linux/Unix-Rechner haben meist unterschiedliche Bezeichnungen für die Sicherungsmedien.

- Für Benutzer, die oft ihre digitalen Bilddaten speichern wollen, empfiehlt es sich, gleich eine zusätzliche externe USB-Platte zu verwenden.
- Bei kleineren Datenmengen reicht auch schon ein USB-Stick
- Mit dem Kommando `df -h` können Sie feststellen, wo das Gerät eingehängt wurde und wie viel Platz Ihnen hierauf zur Verfügung steht.

Einbinden von Wechselmedien

Unter Unix müssten Sie meist den USB-Datenträger selbst einbinden mit dem Kommando `mount`. Da evtl. das `mount`-Kommando nicht für »**normale Benutzer**« freigegeben ist, müsste dies u.U. unter der `root` erfolgen, bzw. der Systemverwalter müsste das Kommando für Sie freigeben.

```
mount /dev/Gerätename /Mountpunkt (Verzeichnis)
```

Wichtig: Als Pfad wird der absolute Pfad angegeben: z.B. `/mnt`

Komprimierungstools

Die Sicherung wurde hier durch Kopieren bestimmter Dateien vorgenommen. Könnten wir den Stick nun wieder entfernen?

Nein, erst muss der Stick abgemeldet (sicher entfernt werden).

umount /dev/Gerätename oder /Mountpunkt (Verzeichnis)

Abmontieren, aushängen

Durch das Kommando **umount** ist eine ordnungsgemäße Rückschreibung aller Dateinformationen gewährleistet.

Erstellen einer Sicherung mit tar

Statt wie mit cp die Dateien einzeln zu kopieren, erstellt man besser eine Archivdatei, in der die Daten in ähnlicher Struktur wie auf der Platte abgebildet werden.

Hierbei bleiben im Normalfall die ursprünglichen Eigenschaften der Dateien, wie:

- Datum
- Zugriffsrechte
- etc., erhalten.

Das Kommando tar (tape archive) war ursprünglich dazu gedacht, Daten auf Magnetbänder zu schreiben, was natürlich immer noch möglich wäre, doch die heutigen Medien bieten hier bessere Möglichkeiten.

tar -c - Kommando, um ein tar-Archiv zu erstellen

tar -c[bhvj]f Archivname Verzeichnis oder Dateiname(n)

- c (create) Erstellen eines tar-Archivs
- b (block) Blockungsfaktor (Wenn die Sicherung auf ein Magnetband geschrieben wird)
- h (hardlink) verfolgt alle symbolischen Links und kopiert die dort aufgefundenen Dateien
- v (verbose) zeigt alle kopierten Dateien an
- z die erstellte Archivdatei wird mit gzip komprimiert
- j die erstellte Archivdatei wird zusätzlich mit bzip2 komprimiert
- f das direkt nachfolgende Wort ist die Archivdatei, f wird daher immer als letztes angegeben

Die Optionen in den eckigen Klammern sind optional.

Komprimierungstools

- Um sofort zu erkennen, dass es sich bei der Datei um ein tar-Archiv handelt, sollte als Suffix **.tar** verwendet werden.
- Wurde die Datei gleich mit **z** oder **j** komprimiert, ergänzt man die Datei mit **.tar.gz** bzw. **.tar.bz2**.

```
sb@ub:~$ tar -cvf mein.archiv.tar test/
test/
test/neu
test/fehler
test/projektA
test/befehle
test/ueb1
test/D*
test/Datum
test/inhalt
test/inhalt2
sb@ub:~$
sb@ub:~$ ls -l *.tar
-rw-rw-r-- 1 sb sb 10240 Apr 17 15:09 mein.archiv.tar
```


Extrahieren von Dateien aus einem tar-Archiv

Wichtig beim Restaurieren von Dateien ist, dass Sie sich bewusst sind, dass bei tar die Dateien stets in das aktuelle Directory kopiert werden, und zwar so, wie sie bei der Sicherung mit ihrem relativen Pfadnamen angezeigt wurden.

```
tar -t[vzj]f Archivname [Dateiname(n)]
```

- t (table) Inhaltsverzeichnis
- v (verbose) zeigt alle kopierten Dateien an
- z falls die erstellte Archivdatei mit gzip komprimiert wurde
- j falls die erstellte Archivdatei zusätzlich mit bzip2 komprimiert wurde
- f das direkt nachfolgende Wort ist die Archivdatei, f wird daher immer als letztes angegeben

Werden Dateinamen angegeben, wird nur nach diesen gesucht, und die gefundenen angezeigt.

Extrahieren von Dateien aus einem tar-Archiv

Erst dann sollten Sie Daten wieder einlesen bzw. Daten aus dem Archiv extrahieren, herausholen. (Was allerdings nicht bedeutet, dass die Dateien aus dem Archiv gelöscht werden. Sie werden nur zurückkopiert, bleiben aber in der Archivdatei erhalten.)

```
tar -x[mvzj]f Archivname [Dateiname(n)]
```

- x (extract) Herauskopieren von Dateien und/oder Verzeichnisse
- m (modification date) Übernahme des Originaldatums
- v (verbose) zeigt alle kopierten Dateien an
- z falls die erstellte Archivdatei mit gzip komprimiert wurde
- j falls die erstellte Archivdatei zusätzlich mit bzip2 komprimiert wurde
- f das direkt nachfolgende Wort ist die Archivdatei, f wird daher immer als letztes angegeben

Extrahieren von Dateien aus einem tar-Archiv

Wenn Sie versehentlich den Ordner test/ gelöscht haben, kann dieser auf den folgenden Weg wiederhergestellt werden:

```
sb@ub:~$ rm -rf test/
sb@ub:~$
sb@ub:~$ tar -xvf mein.archiv.tar
test/
test/neu
test/fehler
test/projektA
test/befehle
test/ueb1
test/D*
test/Datum
test/inhalt
test/inhalt2
sb@ub:~$
sb@ub:~$ ls test
befehle  'D*'  Datum  fehler  inhalt  inhalt2  neu  projektA  ueb1
```

Komprimieren von Dateien

Bei **tar** konnte man die Datei gleich mit **-z** oder **-j** komprimieren. Doch auch im Nachhinein ist dies möglich. Um Dateien zu komprimieren, gibt es die Kommandos:

```
gzip [-vdt]Dateiname(n)
```

- v Verbose Modus
- d Extrahiert das Archiv
- t Testet die Integrität des Archivs

Ohne Angabe von Optionen, wird das angegebene Archiv komprimiert

Komprimieren von Dateien

Komprimieren wir nun in unserem Beispiel die Sicherung test.gz mit **gzip**. Die ursprüngliche Datei wird gelöscht und es gibt nur noch die mit **.gz** komprimierte Datei.

- Über **gunzip** wird die ursprüngliche Datei wieder hergestellt:

```
sb@ub:~$ ls -l mein.archiv.*
-rw-rw-r-- 1 sb sb 10240 Apr 17 15:09 mein.archiv.tar
sb@ub:~$
sb@ub:~$ gzip mein.archiv.tar
sb@ub:~$
sb@ub:~$ ls -l mein.archiv.*
-rw-rw-r-- 1 sb sb 540 Apr 17 15:09 mein.archiv.tar.gz
sb@ub:~$
```

Neuer, schneller, besser: „bzip2“

Verwendet einen anderen Kompressionsalgorithmus und verkleinert daher die meisten Dateien deutlich besser.

- Weiterhin besitzt Bzip2 einen so genannten Recovery-Modus: Beim Komprimieren zerlegt das Tool die Dateien in einzelne Blöcke.
- Ist eine Datei beschädigt, ist es eventuell noch möglich, Daten aus den intakt gebliebenen Bereichen zu retten – dazu entpacken Sie mit Bzip2recover die unbeschädigten Teile.

Wann und wie oft sollten Sie sichern?

Hierbei sollten Sie berücksichtigen, dass Sie nicht immer sofort einen evtl. Fehler innerhalb von Dateien oder das Fehlen bestimmter Dateien bemerken.

- Auch ist es bei einem großen Datenbestand aus Zeit- und Platzgründen zu empfehlen, nicht jeden Tag alle Dateien zu sichern, sondern nur noch jene, die sich seit der letzten Sicherung verändert haben oder neu angelegt wurden. Nehmen Sie die Sicherung genau.
- Die Zeit, die Sie vielleicht einsparen, wenn Sie nur von Zeit zu Zeit sichern, könnte Ihnen teuer zu stehen kommen. Die Sicherung sollte zur täglichen Routine werden. Am besten erstellen Sie einen Sicherungsplan

Wann und wie oft sollten Sie sichern?

Eine ideale Lösung wäre, die Sicherung z.B. nachts automatisch über das Netz auf einem eigenen Rechner (File-Server). Hier gibt es unter Linux/Unix Kommandos, mit denen zu bestimmten Zeiten Befehle gestartet werden sollen. Hierzu gehört:

at Zeitangabe [Kommando]

Auszuführende Kommandos können in eigenen Zeilen angegeben werden. Ende mit **Strg+d**


Die Zeitangabe erfolgt mit dem folgenden Muster: HHMM (4 Ziffern)
z. B. **2330** (am selben Tag um 23:30 Uhr)

Es können auch Job an folgenden Tagen ausgeführt werden, mehr dazu mit: man at

Wann und wie oft sollten Sie sichern?

Das at-Kommando ist nicht auf allen Systemen für alle Benutzer verfügbar. Sehen wir uns trotzdem kurz die Funktionen von at in einem Beispiel in der Abbildung an:

Strg + d



```
sb@ub:~$ at 2330
warning: commands will be executed using /bin/sh
at> tar -czf /Sicherung/2020-04.tar /home/*
at> <EOT>
job 1 at Fri Apr 17 23:30:00 2020
sb@ub:~$
```

Wann und wie oft sollten Sie sichern?

Bereits gestartete at-Jobs können nachträglich kontrolliert oder wieder gelöscht werden. Hierzu wird das at-Kommando mit Optionen angegeben:

```
at [-r Jobnummer] [-ld Jobnummer]
```

- r (remove) löscht den angegebenen Job
- l (list) zeigt alle noch vorhandenen at-Jobs an
- d (display) zeigt den Inhalt des at-Jobs an

Aufbewahrungsort:

- Generell sollten Sicherungen auf Datenträger vor Staub, Schmutz, Feuchtigkeit, Feuer und magnetischen Einflüssen geschützt sein und nur für Systemverwalter oder bestimmte Personen zugänglich sein – also gut abgeschlossen z.B. in einem Schrank aufbewahrt werden.
- Für wichtige Dateien empfiehlt es sich, die Sicherungsmedien sogar in einen Tresor zu geben.
- Bei eigenen Produkten (Software, Dokumentation etc.) wäre als Aufbewahrungsort ein Banksafe nicht verkehrt.

Allgemein:

- Wie Sie Ihre Sicherung einrichten, hängt zweifelsohne von der Wichtigkeit und Änderungshäufigkeit des Datenbestandes ab.
- Doch nehmen Sie die Sicherung nicht auf die leichte Schulter, fragen Sie nach, wie das Sicherungskonzept Ihrer Firma bzw. Ihrer Abteilungsrechner aussieht.

dd – disk dump

dd dient zum bit-genauen Kopieren von Festplatten, Partitionen oder Dateien.

- "Bit-genau" Kopieren bedeutet, dass der Datenträger Bit-für-Bit bzw. Byte-für-Byte ausgelesen und beschrieben wird, unabhängig von dessen Inhalt und Belegung.
- dd funktioniert grundsätzlich mit allen Dateisystemen auf die Ubuntu / Linux zugreifen kann (z.B. ext2/3, reiserfs, vfat, ntfs etc.).
- Es funktioniert auch mit CD/DVD-Dateisystemen, allerdings nur für Daten-CDs/DVDs.

dd – disk dump

Spiegel von Festplatten / Partitionen

```
dd if=Quelle of=Ziel [Optionen]
```

Beispiel, Partition:

```
dd if=/dev/sda1 of=/dev/sdb1
```

dd – disk dump

Images von Festplatten / Partitionen erstellen:

```
dd if=Quelle of=Ziel.img [Optionen]
```

Beispiel, Partition:

```
dd if=/dev/sda1 of=/data/backup.img
```



**VIELEN DANK
FÜR IHRE
AUFMERKSAMKEIT!**