

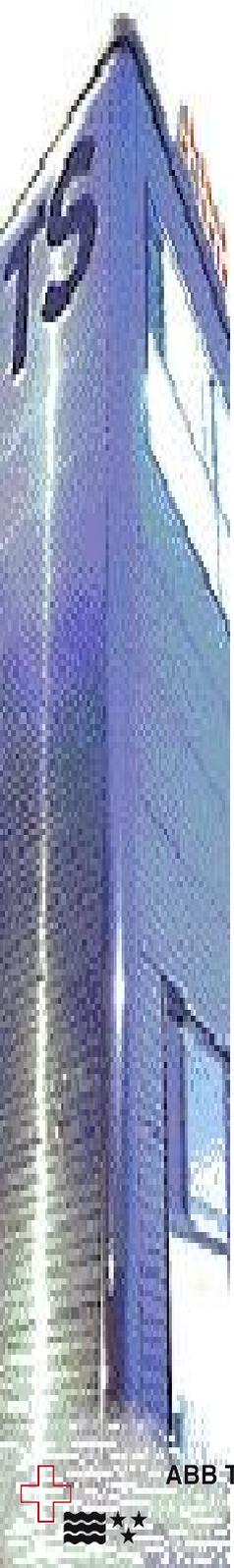
# Datensicherung und Archivierung

---

## Lernziele



- Grundprinzipien der Datensicherung
- Einige Linux-Werkzeuge zur Datensicherung kennenlernen



# Allgemeines

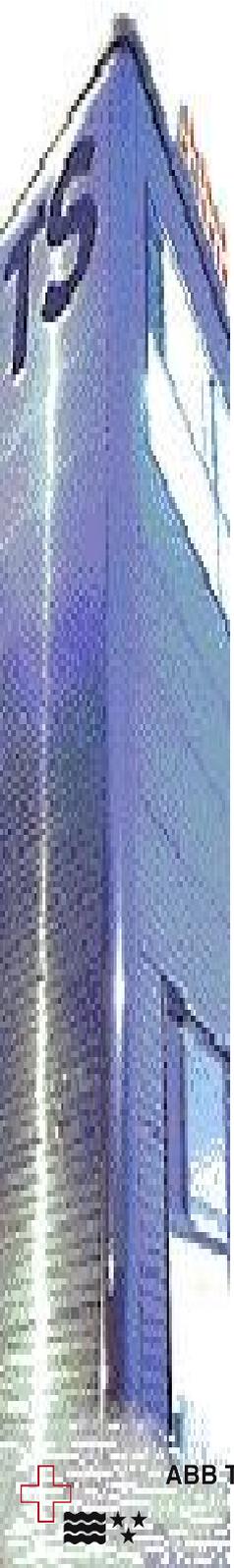
- Es ist die Aufgabe eines Systemadministrators, die im Laufe der Zeit anfallenden Daten regelmässig zu sichern
- Es gibt zahlreiche Situationen, die zu einem Datenverlust führen können
  - versehentliches Löschen
  - Hardwarefehler
  - Fehler des Betriebssystems

# Allgemeines

- Es gibt verschiedenste Ansätze für eine Datensicherung
- Es können unterschiedliche Medien verwendet werden
  - Kopie auf ein anderes Dateisystem / Disk
  - Externe Festplatten
  - CD/DVD Medien
  - Bandlaufwerke

# Allgemeines

- RAID-Systeme, also redundant ausgelegte Plattenspeicher, können die Wahrscheinlichkeit von Plattenausfällen verringern
- Sicherheitskopien werden damit aber nicht überflüssig



# Archivierungsstrategien

- Prinzipiell können zwei Arten der Datensicherung unterschieden werden
- Online-Sicherung
  - wenn ein wichtiger Datenbestand sofort zu sichern ist
- Offline-Sicherung
  - hier wird der Datenbestand zu Zeiten geringerer Systemauslastung, also zumeist nachts, automatisch gespeichert

# Archivierungsstrategien

- Prinzipiell sollte sich anhand der Datenarchive über Wochen oder gar Monate hinweg der Datenbestand eines bestimmten Tages restaurieren lassen
- Es ist nicht sinnvoll, jeden Tag sämtliche Daten erneut zu sichern
- Es reicht aus, nur die seit der letzten Sicherheitskopie veränderten Dateien zu sichern (inkrementelle Archivierung)

# Archivierungsstrategien

- Eine gängige Strategie ist es, einmal pro Woche eine Komplettsicherung zu machen
  - Full Dump, Stufe 0
  - Danach werden nur noch die Daten seit der Komplettsicherung erfasst
    - Incremental Dump, Stufe 1
    - Dann die veränderten Daten ab Stufe 1
      - Incremental Dump, Stufe 2
      - USW...

# Archivierungsstrategien

- Ein Faktor, der oftmals vergessen wird, ist das ausgiebige Testen der Archivierungen
- Ein Backup Medium sollte zwischendurch einem Test unterzogen werden.
- Für Server-Systeme sollte einmal ein Disasterrecovery durchgespielt werden
- Logdateien und Statusmeldungen müssen beachtet werden

# Übungen

- [9.1] Was versteht man unter dem Begriff »inkrementelle Sicherung«?

# Übungen

- [9.2] Diskutieren Sie die Vor- und Nachteile von Bandgeräten als Archivierungslaufwerke.

# Archivierungslaufwerke

- Auf den meisten Systemen werden bevorzugt Magnetbänder zur Datensicherung eingesetzt
- Magnetbänder haben einen sequentiellen Zugriff
- Die Daten auf den Medien werden blockweise abgelegt
- Bandlaufwerke werden aber als zeichenorientierte Geräte angesprochen

# Archivierungslaufwerke

- Magnetbänder müssen normalerweise nicht wie Festplatten formatiert werden
- Es gibt kein Dateisystem auf dem Band
- Das Einbinden der Medien mit mount ist nicht vorgesehen
- Die Daten werden als kontinuierlicher Strom auf das Medium geschrieben

# Archivierungslaufwerke

- Bandlaufwerke können wahlweise als »zurückspulende« (engl. rewinding) oder »nichtzurückspulende« (engl. non-rewinding) Geräte angesprochen werden
- Die Auswahl einer Betriebsart erfolgt anhand der Gerätedateien
  - # /dev/st0 zurückspulendes Gerät
  - # /dev/nst0 nicht zurückspulendes Gerät

# Archivierungslaufwerke

- Archivierungsdaten werden als kontinuierlicher Datenstrom auf ein Band geschrieben
- Sie verhalten sich wie eine grosse Datei
- Am Ende wird auf dem Band eine Markierung für das Dateiende geschrieben (engl. end of file, kurz EOF)
- Auf das Band können weitere Datenströme geschrieben werden

# Archivierungslaufwerke

- Mit dem Kommando *mt* kann das Band hinter der Endmarkierung positioniert werden
- ```
# mt -f /dev/nst0 fsf 1
```
- Diese Operation muss immer im »non-rewinding«-Modus ausgeführt werden
- Die Operation *fsf 1* (engl. forward skip file) spult das Band bis zur ersten Dateiendemarke vor

# Arbeiten mit Bandlaufwerken

- Das Programm *mt* stellt die Grundoperationen von Bandlaufwerken bereit
- mt* benutzt das Gerät */dev/tape*
- Es kann über die Variable *TAPE*
  - oder mit der Option *-f* geändert werden
- Auch ein entferntes Gerät kann über das Netz angesprochen werden
  - # `hugo@tapeserver.example.com:/dev/nst1`

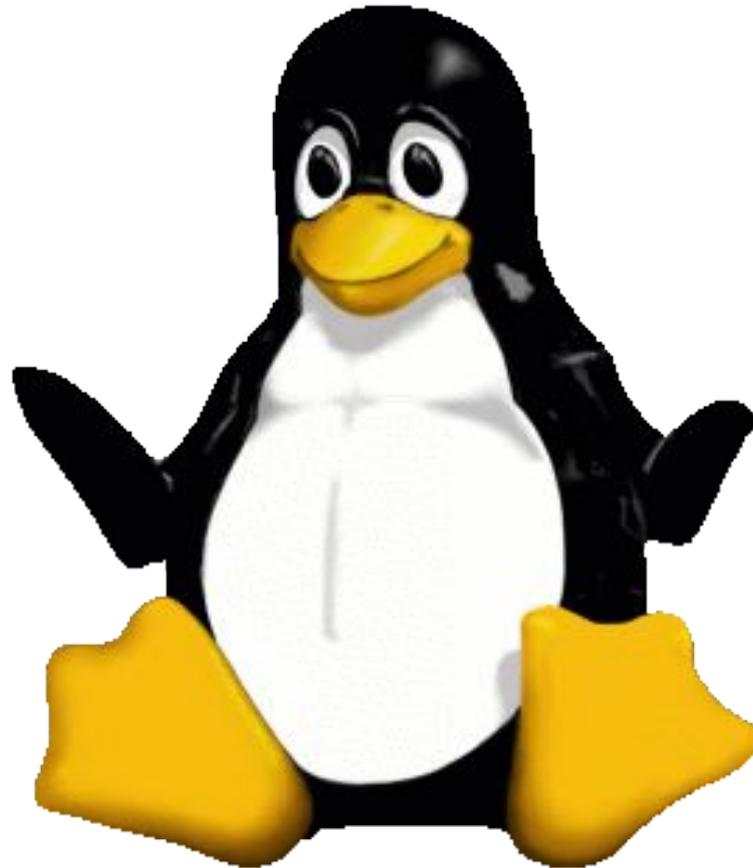
# Arbeiten mit Bandlaufwerken

# mt [<Option> ...] <Operation>

|              |                                                     |
|--------------|-----------------------------------------------------|
| eof, weof    | schreibt EOF-Marken                                 |
| fsf <i>n</i> | spult <i>n</i> Dateien vor                          |
| bsf <i>n</i> | spult <i>n</i> Dateien zurück                       |
| asf <i>n</i> | Absolute Positionierung zu Datei <i>n</i>           |
| eom          | spult das Band ans Ende des<br>bespielten Bereiches |
| rewind       | spult das Band zurück                               |
| status       | gibt den aktuellen Zustand aus                      |
| erase        | löscht das Band                                     |
| offline,     |                                                     |
| rewoffl      | wirft das Band aus                                  |

# Fragen?

---



# Übungen

- [9.3] Mit welchem Programm können Sie Magnetbänder steuern?
- Wie müsste der Befehl lauten, um das Band im ersten SCSI-Magnetbandgerät an den Bandanfang zurückzuspulen?

# Sichern im Netzwerk

- Kommerzielle Programme
  - IBM Tivoli Storage Manager
  - Legato
  - Backup EXEC, uva...
- Open Source
  - Amanda
  - Bacula
  - rsync
  - Eignet sich für Spiegelung der Daten



# Archivierungsprogramme

- Datensicherung mit *tar* (*tape archive*)
- Dateien und ganze Verzeichnisbäume werden mit Zusatzinformationen (Datum, Zugriffsrechte, Eigentümer, . . . ) in eine Archivdatei gepackt
- Standard für die Verbreitung von Linux-Programmquellcodes
- Die Anwendung ist einfach und es ist zuverlässig

# Archivierungsprogramme

# tar <Optionen> <Datei> | <Verzeichnis> ...

- c erzeugt ein neues Archiv
- f <Datei> erzeugt oder liest ein Archiv aus <Datei>. Kann auch ein Gerät sein
- t zeigt den Inhalt des Archivs
- v ausführlicher Modus, zeigt was läuft
- x Auslesen der gesicherten Dateien
- z komprimiert/dekomprimiert mit gzip
- j komprimiert/dekomprimiert mit bzip2

! tar sichert automatisch relative Pfadnamen

# Archivierungsprogramme

## Ein paar Beispiele:

```
# tar -cvf ~/data.tar data*
```

```
# tar -cvf /dev/nst0 data*
```

```
# cd /; tar -cvf home.tar /home
```

```
# tar -tvf home.tar
```

```
# tar -cvf root@archiv:/dev/tape data*
```

```
  --rsh-command=ssh
```

```
# tar -cf - / | (cd /mnt; tar -xf -)
```

# Archivierungsprogramme

- Datensicherung mit *cpio*
- *cpio*-Archive sind ähnlich den *tar*-Archiven
- *cpio* kennt verschiedene Archivformate
  - binary, crc
  - old ASCII, new ASCII
  - HPUX binary, HPUX old ASCII
  - old tar, and POSIX.1 tar
- Es eignet sich für den Austausch von Daten zwischen verschiedenen Systemen

# Archivierungsprogramme

- *cpio* speichert effizienter als *tar*
- Fehlerhafte Stellen werden übersprungen
- *cpio* arbeitet mit den Standard Ein- und Ausgaben
- Für die Ausgabe von Dateien in ein Archiv eignet sich *find* mit der Option *-depth*
- Aus einem Archiv können Daten über reguläre Ausdrücke gefiltert werden

# Archivierungsprogramme

## Die Syntax ist:

# `cpio -o <Option> ...`

# `cpio -i <Option> <Muster>`

# `cpio -p <Option> <Verzeichnis>`

## Ein Backup in eine Datei:

# `find / -maxdepth 1 -name '*' | cpio -ov >data-all`

## Die Datei wird entpackt mit dem Befehl:

# `cpio -iv <data-all`

# Archivierungsprogramme

- Das klassische Unix-Programm zur Datensicherung ist *dump*
- *dump* greift direkt auf die Datenstrukturen des Dateisystems zu
- Daher ist *dump* dateisystemspezifisch
- Dateisysteme sollten nicht gemountet sein
- ! Linus Torvalds nennt *dump* als Backup “russisches Roulette“
- <http://lwn.net/2001/0503/a/lt-dump.php3>

# Archivierungsprogramme

Die Syntax ist:

```
# dump -<Stufe> -f <Laufwerk> <Dateisystem>
```

Ein Backup auf Band:

```
# dump -0 -f /dev/st0 /home
```

Die <Stufe> bestimmt den Typ der Sicherung

0 entspricht einer kompletten Sicherung

1-9 sind Inkrement der jeweils höheren Ebene

# Archivierungsprogramme

- Das Gegenstück zu *dump* ist *restore*
- *restore* unterstützt mehrere Betriebsarten
  - Interaktives Restaurieren von Dateien
    - # `restore -i -f /dev/st0`
    - Kommandos wie *cd*, *ls*, *add*, *delete* oder *extract*
  - Automatisches Restaurieren
    - # `restore -rf /dev/st0`
    - Ebene-0-Sicherung einspielen, Ebene-1 usw.
- Prüfmodus
  - Mit der Option *-C* wird ein Vergleich gemacht

# Archivierungsprogramme

- Partitionen sichern mit *dd*
- *dd* ist nicht im eigentlichen Sinne ein Archivierungsprogramm
- Eignet sich aber für ein schnelles Disaster Recovery
- Daten werden bitweise aus einer Eingabedatei gelesen (keine Dateien)
- Die Art der Daten spielt keine Rolle

# Archivierungsprogramme

● Die Syntax ist:

● # `dd if=/data/sda of=/dev/sdb`

● Die Platten müssen die gleiche Architektur haben

● *dd* kann auch den Inhalt von CDRROMs oder DVDs in eine Datei schreiben

● # `dd if=/dev/cdrom of=/data/cdrom1.iso`

● Das ISO-Image lässt sich dann mounten

● # `mount -t iso9660 -o loop,ro /data/cdrom1.iso /mnt`

# Archivierungsprogramme

- Der Nachteil bei *dd* ist, dass immer alle Datenblöcke gesichert werden
- Das Programm *partimage* kann mit Dateisystemen umgehen und sichert nur belegte Blöcke
- Partimage kann auch Partitionen ändern und verschieben
- Mit Partimage können Images auf andere Hardware installiert werden

# Datenkomprimierung

- Ursprünglich wurde unter UNIX das Programm *compress* verwendet
- Compress erzeugt Dateien mit der Endung “.Z“
- Zum entpacken wird *uncompress* verwendet.
- Archive können mit *zcat* ausgegeben werden
- Z.B. um mit *grep* nach einem Muster zu suchen

# Datenkomprimierung

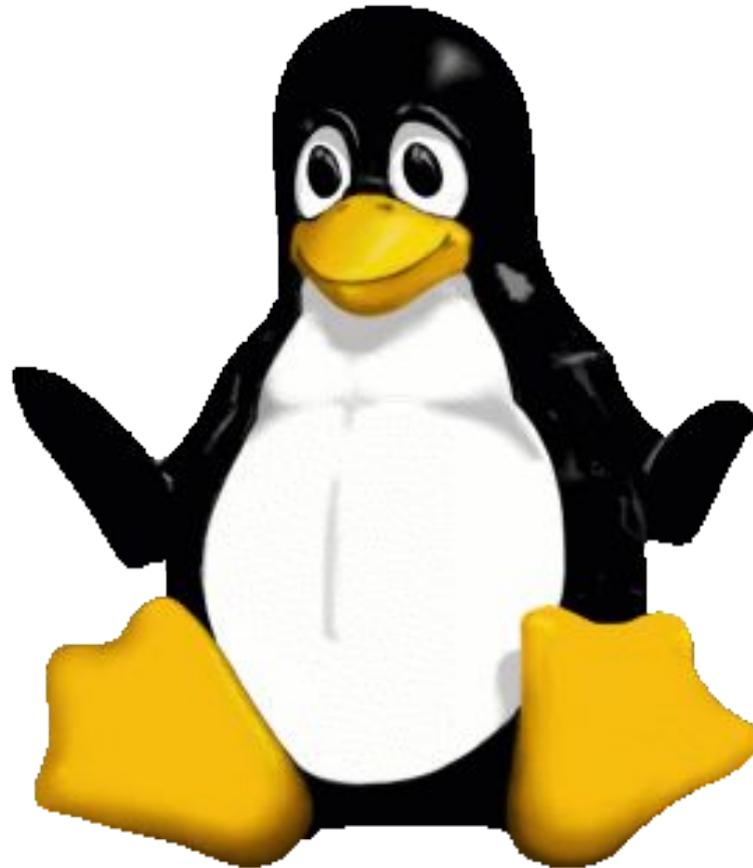
- Unter Linux wird zum komprimieren meist *gzip* (GNUzip) verwendet
- *gzip* verwendet die Datei-Endung “.gz“
- *gzip*-Dateien können mit Zip-Programmen von anderen Systemem gelesen werden
- Sollen mehrere Dateien komprimiert werden, können *tar* und *gzip* kombiniert werden
- Archive kann man mit *gunzip* entpacken

# Datenkomprimierung

- Der folgende Befehl komprimiert die Datei `brief.tex`
  - `# gzip brief.tex`
- Entpackt wird mit:
  - `# gzip -d brief.tex`  
oder
  - `# gunzip brief.tex`
- Mit der Option `-s` kann eine von `“.gz“` abweichende Endung angegeben werden

# Fragen?

---



# Übungen

- [9.4] Legen Sie das Inhaltsverzeichnis Ihres Heimatomdners in eine Datei *inhalt* ab.
- Erzeugen Sie aus dieser Datei ein tar-Archiv.
- Betrachten Sie die Ursprungsdatei und das Archiv.
- Was bemerken Sie?

# Übungen

-  [9.5] Legen Sie drei bis vier leere Dateien an und fügen Sie diese dem gerade erzeugten Archiv hinzu.

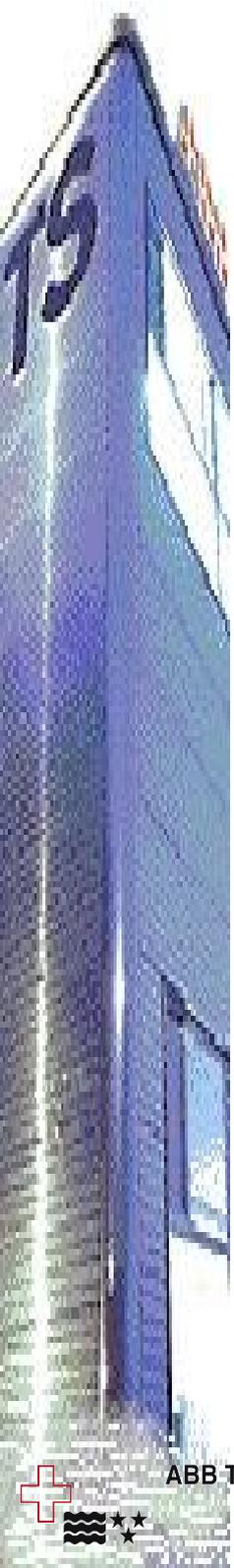


# Übungen

- [9.6] Löschen Sie die Originaldateien und entpacken Sie danach den Inhalt des tar-Archivs.

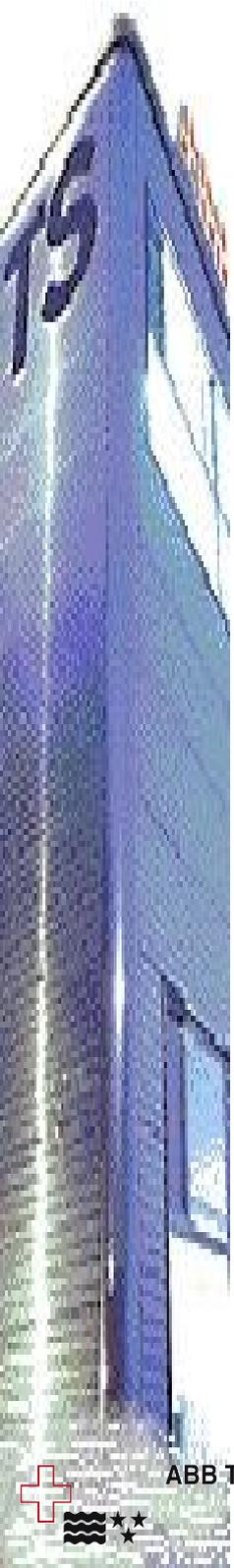
# Übungen

- [9.7] Komprimieren Sie das tar-Archiv mit maximaler Verkleinerung.



# Übungen

- [9.8] Sehen Sie sich den Inhalt des komprimierten Archivs an.
- Stellen Sie das ursprüngliche tar-Archiv wieder her.



# Übungen

- [9.9] Wie gehen Sie vor, wenn Sie den gesamten Inhalt Ihres Heimatverzeichnis in eine mit gzip komprimierte Datei packen möchten?

# Übungen

- [9.10] Stellen Sie sich vor, Sie machen regelmässige Sicherheitskopien nach dem oben erklärten modifizierten Türme-von-Hanoi-Schema.
- In der Nacht von Mittwoch auf Donnerstag geht Ihre Festplatte kaputt.
- Welche Sicherungskopien müssen Sie einspielen?

# Übungen

- [9.11] Sie können mit dd ein komplettes »Image« einer Platte anlegen.
- Zurückspielen sollten Sie es nach Möglichkeit jedoch nur auf eine baugleiche Platte.
- Warum?