Heinz-Nixdorf-Berufskolleg

für Elektrotechnik, Informations- und Telekommunikationstechnik der Stadt Essen

Archivierung und Backup unter Linux mit tar

Diese Übung ist auf Ubuntu-Linux zugeschnitten, sollte aber auch unter anderen Linux-Systemen funktionieren.

Das Pack-Tool `tar` darf sicherlich als Dinosaurier unter den Packern bezeichnet werden. Der Name leitet sich aus *Tape Archiver* ab. Allerdings kann `tar` seine Ausgaben auf alle möglichen Speichergeräte ablegen.

Aufruf:
`tar [Optionen] <Archiv> <Dateien/Verzeichnisse>`

Aufstellung der wichtigsten Optionen:

- `-c create`: erzeugt ein neues Archiv
- `-f Archiv-Datei` (Name)
- `-r append`: Dateien an das Archiv anhängen
- `-t` listet den Inhalt des Archiv nur auf
- `-x extract`: entpackt Dateien aus dem Archiv
- `-z` Archiv durch gzip filtern (komprimieren, dekomprimieren)
- `--exclude=<Datei>` Datei <Datei> ausschließen

Weitere Hilfe und Informationen erhält man mit `tar --help` bzw. den Manualpages `man tar`. `tar` arbeitet mit relativen Pfaden, daher auch häufig die Ausgabe `...` `removing leading /` ...

Übung 1: Backup des /etc-Ordners

- einloggen als `root` und – wenn nicht schon automatisch erfolgt – wechseln in das Heimatverzeichnis von root: `/root`
- Packen des /etc-Ordners: `tar -cf etc.tar /etc`
- Kontrolle des Archivs z.B. mittels mc [midnight commander: apt-get install mc]
- Packen und Komprimieren des /etc-Ordners: `tar -czf etc.tgz /etc` Hierbei zusätzlich die Option z verwenden und dem Archiv die Endung tgz (tar – gzip) vergeben!
- Vergleichen Sie die Größen der in dieser Übung erzeugten Dateien...
- Sie können ein tar-Archiv auch „zu Fuß" komprimieren: `gzip etc.tar`
- Beachten Sie die neue Dateiendung!

Scanned by CamScanner

Übung 2: Anzeigen von Archiv-Inhalten

Verwenden Sie den Parameter t: tar -tf <archiv-name> optional mit Pipe nach less:
`tar -tf <archiv-name> | less`

Übung 3: Einfaches inkrementelles Backup mit tar

Um das Alter von Dateien herauszufinden, verwenden wir den Befehl find mit den Parametern `-mtime` bzw. `-mmin`

Beispiel: Erzeugen Sie mittels `touch /etc/test.txt` eine neue Datei im /etc Ordner und geben Sie anschließend `find /etc -mmin -1` ein. Jetzt sollte die neu erstellte Datei angezeigt werden.

`find /etc -mmin -1` liefert die Dateien im /etc-Ordner, die neuer sind als 1 Minute
`find /etc -mmin -5` liefert die Dateien im /etc-Ordner, die neuer sind als 5 Minuten
`find /etc -mmin +5` liefert die Dateien im /etc-Ordner, die **älter** sind als 5 Minuten
`find /etc -mtime -1` liefert die Dateien im /etc-Ordner, die neuer sind als 1 **Tag**

Um die aufgelisteten neuen Dateien zu sichern, wird der Umweg über eine temporäre Datei (`neu.txt`) gewählt:

- `find /etc -mmin -1 > neu.txt` sichert die Liste der neuen Dateien, die jünger als 1 Minute sind in der Datei `neu.txt`.
- `tar -cf inc_etc.tar -T neu.txt` sichert nun die in der Datei `neu.txt` angegebenen Datei in das Archiv `inc_etc.tar`.

Einfache Skripte hierzu könnten folgendermaßen aussehen:

```
vollbackup.sh:
cd /root
#Vollbackup von /test
tar -cf /root/test.tar /test
#Komprimieren
gzip /root/test.tar
inkrementell.sh:
Date=`/bin/date +"%d%m%Y-%H%M%S"`
cd /root
#Was ist in der letzten Minute geändert worden?
#Findest alles, was neuer als jetzt -1min ist.
find /test -mmin -1 > neu.txt
#Packen
tar -cf /root/inc_test.$Date.tar -T /root/neu.txt
rm /root/neu.txt
```

- erzeugen Sie die obigen Skripte im Ordner `/root`
- legen Sie nun den Ordner `/test` an
- erzeugen Sie im Ordner `/test` 2 beliebige Dateien
- starten Sie das Skript für das Vollbackup (in /root): `sh vollbackup.sh`
- warten sie gut zwei Minuten!
- erzeugen Sie eine neue Datei in `/test: touch neuneu.txt`
- starten sie sofort danach wiederum das Skript für das inkrementelle Backup und untersuchen Sie die erzeugten Archive!

Scanned by CamScanner

Das Beispiel aus Übung 3 ist eher etwas für den Labortisch. Um ernsthafte inkrementelle Backups zu machen, sollte der Parameter `-mtime` beim find-Befehl verwendet werden. Somit lassen sich z.B. alle 24h die wichtigsten Ordner (z.B. /etc, /usr, /var oder /home) inkrementell sichern. `-mtime -n` bedeutet alles was älter als 24h * n ist.

Um ein differentielles Backup zu realisieren, behält man einfach die Bezugszeit bei einem erneuten Aufruf des Skriptes bei! D.h. Bei täglichen Backups muss der Parameter n nur täglich hochgezählt werden: Mo: 1, Di: 2, ...)
Ganz bequem funktioniert dieser Mechanismus zeitgesteuert. Hier bedienen wir uns des CRON-Dämons.

Übung 4: Cron-Jobs einrichten

`crontab -e` gibt Ihnen die Möglichkeit, Ihre zeitgesteuerten Aufgaben zu verwalten.
Fügen Sie folgende Zeile ein: `* * * * * sh /root/inkrementell.sh`
Beobachten Sie, wie jede Minute ein neues Archiv in `/root` hinzukommt.
Erzeugen Sie gelegentlich neue Dateien im Ordner `/test` oder ändern Sie den Zeitstempel von Dateien in dem Ordner durch den Befehl `touch </test/Dateiname>`

crontab [1]dient der Verwaltung von cron-Tabellen, den Anwenderschnittstellen zum cron-Dämon. Cron wird für das regelmäßige Ausführen bestimmter - mittels crontab - erstellter Arbeitslisten eingesetzt. crontab hat im Wesentlichen drei Aufgaben:

1. Zeigt die aktuelle cron-Tabelle an
2. Erlaubt deren Editieren
3. Wird zum Löschen verwendet.

Für den Systemverwalter ermöglicht crontab den einfachen Zugriff auf bestimmte interne cron-Tabellen, z.B. updatedb.

Aufbau der cron-Tabellen:
cron-Tabellen bestehen aus beliebig vielen Zeilen. Kommentare werden durch # in Spalte eins eingeleitet. Sie gelten bis zum Zeilenende. Leerzeilen werden ignoriert.
Nach der Deklaration lokaler Variablen folgt die eigentliche Tabelle. Diese besteht aus sieben Spalten pro Zeile. Die ersten fünf Spalten spezifizieren den Ausführungszeitpunkt. Die sechste Spalte enthält den Namen des Users, unter dessen Account der Dienst ausgeführt wird. Der Rest wird als siebte Spalte interpretiert und enthält die auszuführenden Befehle.
Die Terminfelder:
Das erste Feld beinhaltet die Minute, das zweite die Stunde, das dritte den Tag, das vierte den Monat. dow (day of the week) steht für den Wochentag.
Innerhalb der Terminfelder können die Angaben als Zahlen (im Feld month und dow auch über die englischen Namen) erfolgen. Mehrere Einzeleinträge sind möglich. Sie werden durch Kommata getrennt. Auch Bereiche können angegeben werden. 2,3,4,6 kann auch als 2-4,6 angegeben werden.
Das *-Zeichen wird als Joker eingesetzt, das den Bereich ersetzen kann (<erstes Element>-<letztes Element>). Die Felder werden durch mindestens ein Leerzeichen oder TAB voneinander getrennt.
Zusätzlich sind noch Zeitschritte (steps) erlaubt: Die Angabe eines /-Zeichens, gefolgt von einer Zahl, bezeichnet eine schrittweise Angabe: */2 bedeutet jeden zweiten Wert, im Feld month also 2,4,6,8,10,12 , im Feld dow entsprechend TU,TH,SA (Di,Do,Sa). 35-49/3 liefert z.B. 35,38,41,44,47.

[1] *Quelle: http://www.linuxhilfen.org/befehle/crontab.html*

Scanned by CamScanner

Beispielaufbau der cron-Tabelle:

```
# (Cron version -- $ID: ... 1.5 2000/05/27 12:05:33 work Exp $)
SHELL=/bin/sh
PATH=/usr/bin:/usr/sbin:/sbin:/bin
#min hour day month dow user command
# Command jede Minute ausführen:
* * * * * root <todo>
# Jeden Tag um 22.00 h alles im Verzeichnis /public
# incl. Unterverzeichnisse löschen:
0 22 * * * pingu rm -r /public/*
# delpub jeden Sa. um 22:00 h aufrufen
0 22 * * 6 root /root/bin/delpub
# Befehl oder Script alle 5 min ausführen
*/5 * * * * root <todo>
# mail_poll alle 2 Stunden ausführen
00 */2 * * * root /usr/local/bin/mail_poll
```

Script zum Beispielaufbau der cron-Tabelle:

```
#!/bin/sh
# delete /public dir
rm -rf /public 2&1>/dev/null
mkdir /public
```

Das Script in /root/bin/delpub ablegen und mit dem Ausführungsrecht versehen:
user@sonne> chmod 700 /root/bin/delpub

Danach aufrufen:
user@sonne> crontab -e

Nun im Editor die Beispielzeilen zu delpub in der crontab eintragen. Ab sofort wird jeden Samstag um 2200h das Verzeichnis /public rekursiv gelöscht und anschließend neu angelegt.

Scanned by CamScanner