

Hinweise zu dem unten anhängenden Script

Das Script umfasst genau den Bereich, der grün unterlegt ist. Das Querformat wurde gewählt, damit keine ungewollten Zeilenumbrüche entstehen.

Kopiere diesen Bereich in einen einfachen Texteditor wie vi, ed, gedit, kedit oder xfwrite und nehme die notwendigen Anpassungen vor. Benutze dafür kein Textverarbeitungsprogramm wie LibreOffice!

Zeilen, die mit einer „#“ beginnen, sind Kommentare. Sie werden ebenso wie leere Zeilen von der bash ignoriert.

Damit das Script fehlerfrei durchläuft, müssen die rot markierten Bereiche an die eigenen Gegebenheiten angepasst werden.

Die gelb unterlegten Bereiche sind optional und können auskommentiert oder geändert werden.

Speichere die Datei zunächst in deinem /home/<Benutzer> - Verzeichnis zum Beispiel mit dem Namen „mein_backup“, und führe in einer root-Konsole folgende Befehle aus:

```
chgrp root /home/<Benutzer>/mein_backup  
chmod 750 /home/<Benutzer>/mein_backup  
/home/<Benutzer>/mein_backup
```

Der 1. Befehl ändert die Gruppe zu root. Der 2. Befehl macht das Script für den Eigentümer und die Gruppe root ausführbar. Der 3. Befehl führt das Script aus.

Im Anschluss an den 3. Befehl erscheinen in der Konsole fortlaufend alle Dateien und Verzeichnisse, die dem Backuparchiv hinzugefügt werden. Die Ausgabe ist hilfreich um zu überprüfen, ob die Angaben in der Datei „ohne_voll.txt“ die gewünschte Wirkung entfalten. Wenn alles richtig funktioniert, sollte das „v“ aus der Option „-cvjp“ der folgenden Zeile im Script entfernt werden.

```
tar -cvjp -X "$OHNE_V" -f "$ZIEL"voll_`date +%Y%m%d`.tar.bz2 "$QUELLE" ;
```

Nun ist alles fertig und das Script wird an die richtige Stelle verschoben und die root-Konsole geschlossen.

```
mv /home/<Benutzer>/mein_backup /etc/cron.weekly/
```

Glückwunsch. Ab sofort läuft im Hintergrund ein automatisches Backup deiner privaten Daten.

```
#!/bin/bash

# Backup Skript für das Verzeichnis /home/<BENUTZER>
# Standard Verwendung in /etc/cron.weekly
#
# Das Script erstellt beim ersten Durchlauf ein Vollbackup,
# danach 12 inkrementelle Backup und dann wieder ein Vollbackup.
#
# Die Variablen ZIEL und QUELLE sind auf die eigenen
# Gegebenheiten anzupassen. Der Verzeichnisname muss einen abschließenden "/" haben!
#
# Bitte nicht vergessen dem chown-Befehl am Ende des Skriptes den richtigen Benutzer
# mitzugeben.

# Beginn Zielpartition einhängen
# Wenn nicht benötigt, die folgenden Zeilen auskommentieren.
# Beachte auch den Abschnitt "Zielpartition aushängen" am Ende des Script.
mount /dev/<DEINE_ZIELPARTITION>
sleep 3
# Ende Zielpartition einhängen

# Das zu sichernde Verzeichnis
QUELLE=/home/<DEIN_BENUTZERNAME>/

# Der Pfad, in dem das Backup und die Hilfsdateien liegen sollen
ZIEL=/<DEIN_ZIELVERZEICHNIS>/

# Auszuschließende Verzeichnisse
#
# Bei Bedarf die Datei 'ohne_inkrement.txt' und 'ohne_voll.txt' im Zielverzeichnis
# erstellen und für jedes Verzeichnis eine Zeile mit dem vollständigen Pfad eintragen.
#
# Achtung: Am Beginn des Pfades kein "/", da der schon in der Variablen "$ZIEL"
# enthalten ist.
# Am Ende der Datei keine Leerzeilen.
#
# Ist die Datei nicht vorhanden oder leer, so wird die Option ignoriert.
OHNE_V="$ZIEL"ohne_voll.txt
OHNE_I="$ZIEL"ohne_inkrement.txt

if [ -f "$ZIEL"datum.txt ] && [ -f "$ZIEL"nummer.txt ] && (( $(< "$ZIEL"nummer.txt) <= 12 )) ;
then
    LETZTE=$(< "$ZIEL"datum.txt)
```

```
ANZAHL=$(  
tar -cjp -X "$OHNE_I" --newer="$LETZTE" -f "$ZIEL"inkrement_`date +%Y%m%d`.tar.bz2 "$QUELLE" ;  
echo $(( ++ANZAHL )) > "$ZIEL"nummer.txt  
  
else  
tar -cvjp -X "$OHNE_V" -f "$ZIEL"voll_`date +%Y%m%d`.tar.bz2 "$QUELLE" ;  
echo 0 > "$ZIEL"nummer.txt  
fi  
  
echo "$(date +%F)" > "$ZIEL"datum.txt  
  
# Dateirechte und Eigentuemer restriktiv setzen.  
chmod 600 "$ZIEL"* 2>/dev/null  
chown <BENUTZERNAME>.<GRUPPENNAME> "$ZIEL"voll* "$ZIEL"inkre* 2>/dev/null  
  
# Beginn Zielpartition aushängen  
# Wenn nicht benötigt, die folgenden Zeilen auskommentieren.  
umount /dev/<DEINE_ZIELPARTITION>  
sleep 3  
# Ende Zielpartition aushängen  
  
exit 0
```