

## Übungsblatt 3: Shellskripte

2. 11. 2012

### Allgemeine Hinweise

- Abgabetermin für die Lösungen ist **Donnerstag, 8. 11., 13:00**.
- Die Lösungen bitte wieder in einer Textdatei speichern, die Du zur Abgabe im Anhang einer Email an Deinen Tutor schickst.

### Aufgabe 3.1: Skripting (5 Punkte)

Betrachte folgendes Skript ( `~arnolda/computergrundlagen/03/script.sh`):

```
#!/bin/bash
# first argument is the command to execute
cmd =$1
# if command is --clean, remove the output from the last run
if [ "$cmd" = "--clean" ]; then
    echo "cleaning up output from last run"
    rm -rf output
    exit 0
if
# bash: prevent the command from being in the @$ expansion
shift
echo "executing $cmd on $@, output goes to directory output"
# create output directory
mkdir -p output
for x in "$@" do
    # remove possible trailing slash (directories + tab-completion)
    x=${x%/}
    # execute and save output
    $cmd $x > output/out-$cmd.${x##*/}.txt
done
```

**Hinweis:** Wie immer gilt: wenn Du einen Befehl nicht kennst, schau unter man nach! bash-interne Befehle wie `while` findest Du unter man bash.

**3.1.1:** Kopiere Dir das obige Skript in Dein Heimatverzeichnis. Was musst Du machen, um diese Datei einfach durch `./script.sh` aufrufen zu können? Gib den entsprechenden Befehl in die Lösungsdatei ein (1 Punkt)

**3.1.2:** Wenn Du das Skript mit

```
script.sh ls ~arnolda/computergrundlagen ~arnolda/computergrundlagen/01
```

ausführst, gibt es noch Fehlermeldungen. Korrigiere diese Fehler, und füge das korrigierte Skript in die Lösungsdatei ein. Füge auch die Ausgabedateien des korrigierten Skripts ein. (2 Punkte)

**Hinweis:** Nicht immer geben die Fehlermeldungen einen direkten Hinweis darauf, an welcher Stelle der eigentliche Fehler liegt. Alle Fehler in diesem Skript können durch das Hinzufügen oder Verschieben von wenigen Zeichen korrigiert werden.

**3.1.3:** Was macht das korrigierte Skript? Beschreibe nicht die einzelnen Befehle, sondern was das gesamte Skript tut. Was geben die Parameter an? (1 Punkt)

**3.1.4:** Warum funktioniert

```
script.sh "ls -l" ~arnolda/computergrundlagen/01
```

nicht? Wie kannst Du das korrigieren? (1 Punkt)

## Aufgabe 3.2: Ein Backupskript (5 Punkte)

Auf dem letzten Übungsblatt hatten wir gesehen, dass sich mit

```
tar -czf backup.tar.gz $HOME
```

ein einfaches Backup realisieren lässt. Leider wird dabei das letzte Backup überschrieben, sofern es an der selben Stelle liegt.

**3.2.1:** Realisiere ein Skript, dass folgendes leistet:

- Die Backup-Dateien werden nach dem Zeitpunkt des Backups in der Form “backup-JJJJ-MM-TT-HH-MM.tar.gz” benannt und in ein eigenes Unterverzeichnis ~/backup gespeichert. Ein am 2. 11. um 13:00 Uhr erzeugtes Backup soll also “backup-2012-11-02-13-00.tar.gz” heißen.
- Existiert das Verzeichnis ~/backup noch nicht, soll es erzeugt werden. Existiert es, ist aber eine Datei, soll es eine Fehlermeldung geben.
- Das Skript soll stets das gesamte Home sichern, egal, von wo aus es aufgerufen wird. Ausnahmen sind die Verzeichnisse ~/backup (klar, dort liegen ja die Backups) ~/.local und ~/.cache (diese können sehr groß werden). Diese Verzeichnisse bitte auslassen.
- Das Skript soll den Benutzer genau über alles informieren, was es tut.

Füge Dein fertiges Skript sowie die Ausgabe eines Probeaufrufs an die Lösungsdatei an! (3 Punkte)

**Hinweise:**

- Das Datum bekommt man mit Hilfe von `date`. Der Backtick (‘), den Du brauchst, um die Ausgabe des Befehls per ‘`date`’ in eine Zeichenkette zu verwandeln, befindet sich auf der englischen Tastatur bei der Tilde (~) links oben.
- Der Übungsleiter kann Fehler nur korrigieren, wenn er verstehen kann, was Deine Intention war. Daher bitte reichlich kommentieren. Ein Kommentar beschreibt *nicht*, was die Befehle tun, sondern, was der Gedanke dahinter ist!
- Beim Testen bitte auf den verfügbaren Speicherplatz (200MB) achten. `du -sh ~` sagt Dir jederzeit, wieviel Platz Du belegst. Eventuell musst Du dann Backups löschen.
- Wenn Du in Schritten vorgehst, ist es einfacher. Probiere zunächst auf dem Terminal, wie Du den Dateinamen des Backups erzeugen kannst, und benutze erst dann `tar`.

**3.2.2:** Ein Backup beugt nicht nur Datenverlusten durch Hardwarefehler vor, sondern hilft auch bei unbeabsichtigtem Löschen. Schreibe daher ein weiteres kleines Skript, das alle Backups nach einer Datei mit gegebenen Namen durchsucht.

Füge Dein fertiges Skript sowie die Ausgabe eines Probeaufrufs an die Lösungsdatei an! (2 Punkte)

**Hinweis:** Eine Möglichkeit ist, eine Schleife über die vorhandenen Backups auszuführen, und mittels `grep` das Inhaltsverzeichnis der Backupdatei nach dem gegebenen Namen zu durchsuchen. Natürlich musst Du dabei auch die Namen der durchsuchten Backup-Dateien ausgeben.