

Linux Advanced

– Aufgaben –

Hinweise

- <> sind Platzhalter, die durch die individuellen Daten zu ersetzen sind.
- X bei den IP-Adressen steht für die Etage
 - UG: 1
 - EG: 2
 - 1.OG: 3
 - 2.OG: 4
 - 3.OG: 5

Y bei den IP-Adressen steht für der Raum (ohne die 0)

BSP.: 10.XY.0.236

Raum 2.04 wird zu 10.44.0.236

Raum E.01 wird zu 10.21.0.236

- `Befehl`
- `Benutzer`
- `Benutzergruppe`
- `Passwort`
- `Dateien oder Verzeichnisse`

1 Aufgabe - Prozesssteuerung

Mit dem Befehl `kill` können sie unterschiedliche Signale an Prozesse senden. Welche gibt es und was machen diese?

Tipp: Es gibt eine vollständige Liste in den Manpages.

2 Aufgabe - cronjob, at, Timer Units

Sie finden unter `/opt/sepe/aufgaben/` unterschiedliche Skripte, die zum bearbeiten der folgenden Aufgaben benötigt werden. Bitte beachten Sie, das ggf. die **Dateiberechtigungen** angepasst werden müssen.

2.1

Das Skript `a.sh` soll einmalig morgen um 8:00 Uhr ausgeführt werden. Wie und mit welchem Tool realisieren Sie dies? Setzen Sie Ihre Lösung entsprechend um.

2.2

Das Skript `b.sh` soll täglich um 7:50 Uhr ausgeführt werden. Wie und mit welchem Tool realisieren Sie dies? Setzen Sie Ihre Lösung entsprechend um.

2.3

Erstellen Sie einen cronjob, welcher das Skript `c.sh` einmal in der Stunde ausführt und die Ausgabe an die Datei `/opt/aufgaben/log/c1.log` anhängt. Bitte beachten Sie, dass das Verzeichnis für die log-Datei erst angelegt werden muss!

2.4

Es gibt bei cronjobs sogenannte Nicknames bspw. `@yearly`. Welche weiteren Nicknames gibt es und wie sieht die cron Schreibweise dafür aus?

2.5

Erstellen Sie eine Timer Unit, die das Skript `e.sh` Dienstag, Mittwoch und Donnerstag um 9:00 Uhr ausführt.

2.6

Es gibt auch bei Timer Units, ähnlich wie bei cronjobs, Nicknames bspw. `yearly`. Ermitteln Sie auch hier welche es gibt und wie die Timer Unit Schreibweise dafür aussieht.

3 Aufgabe - Shell-Programmierung

Vorworte

Alle Ihre Aufgaben sollten neben der Shebang-Zeile folgenden Kommentarkopf enthalten und mit den nötigen Informationen ergänzt werden:

```
#####  
# Name:  
# Klasse:  
# Erstelldatum:  
# Aufgabe:  
# Beschreibung:  
#####
```

3.1

Erstellen Sie ein Shell-Skript, welches überprüft, ob

- das Verzeichnis `/opt/abcd` vorhanden ist.
- die Datei `/opt/efgh.txt` vorhanden ist.
- die Datei `/opt/sepe/aufgaben/a.sh` ausführbar ist.
- die Datei `/opt/sepe/aufgaben/b.sh` neuer als `/opt/sepe/aufgaben/c.sh` ist.

Des Weiteren soll Ihr Skript eine Ausgabe liefern (passend zum Ergebnis; die Gestaltung obliegt Ihnen):

```
#####  
Aufgabe a)  
/opt/abcd: ist (nicht) da  
/opt/efgh.txt: ist (nicht) da  
/opt/sepe/aufgaben/a.sh: ist (nicht) ausfuehrbar  
b.sh ist (nicht) neuer als c.sh  
#####
```

3.2

Erstellen Sie ein Shell-Skript, das drei Variablen enthält und eine Liste erzeugt. Die drei Variablen sind einmal der Start-Wert, End-Wert, Schritte.

Beispiel Ausgabe:

```
#####  
Aufgabe b)  
Start-Wert: 3 End-Wert: 50 Schritte: 5  
#####  
3  
8  
13  
18  
23  
28  
33  
38  
43  
48
```

3.3

Erstellen Sie ein Shell-Skript, das den aktuellen Zeitstempel im folgenden Format ausgeben soll:
dd.MM.YYYY hh:mm:ss.

Beispiel Ausgabe:

```
#####  
Aufgabe c)  
01.04.2021 12:25:33  
#####
```

3.4

Erstellen Sie ein Shell-Skript, das die vier Grundrechenarten beherrscht und bei Übergabe von zwei Parametern alle vier Ergebnisse ausgibt.

Beispiel Ausgabe:

```
#####  
Aufgabe d)  
Eingabe 1: 20  
Eingabe 2: 10  
20+10=30  
20-10=10  
20/10=2  
20*10=200  
#####
```

3.5

Im Verzeichnis `/opt/sepe/aufgaben/` finden Sie eine `user.lst`. Erstellen Sie ein Shell-Skript, welches die Liste einliest und den Benutzer auf Ihrem System anlegt. In der Liste sind auch die Passwörter nach dem „:“ angegeben. Dies sollen Sie ebenfalls mit setzen.

Testen Sie, nachdem Sie das Skript ausgeführt haben, ob Sie sich mit den Benutzern und dem jeweiligen Passwort anmelden können.

Beim `useradd` Kommando müssen Sie einen Passworthash angeben. Diesen erzeugen Sie mit `openssl passwd -1 <password>`

Beispiel Ausgabe:

```
#####
Aufgabe e)
folgende Benutzer wurden erstellt :
Benutzer          Passwort
-----
Max               ABCDE
Klaus             password
...
#####
```

3.6

Erstellen Sie ein Shell-Skript, welches zwei Parameter als Eingabe (mittels `read`) bekommt. Der erste Parameter gibt den Start-Wert an; der zweite den End-Wert. Ihr Skript soll dann unter `/opt/aufgaben/shell/f/` ein Verzeichnis anlegen, welches den Zeitstempel enthält.

Zum Beispiel: `/opt/aufgaben/shell/f/20210413111213`

Danach soll Ihr Skript die Dateien von Ihrem Start-Wert bis zum End-Wert wie folgt anlegen:

Start-Wert: 5 End-Wert: 12

0005.t.tmp, 0006.t.tmp, 0007.t.tmp, 0008.t.tmp, 0009.t.tmp, 0010.t.tmp, 0011.t.tmp

Des Weiteren soll Ihr Skript folgende Ausgabe liefern (passend zu Ihrer Eingabe):

```
#####
Aufgabe f)
Start-Wert: 5
End-Wert: 12
Dateien angelegt in: /opt/aufgaben/shell/f/20210413111213
#####
```

Zusätzlich soll Ihr Skript Fehleingaben abfangen und eine Beschreibung ausgeben, wenn zu wenig bzw. zu viele Parameter übergeben wurden, wie Ihr Skript verwendet werden kann. Sie müssen nicht überprüfen, ob Zahlen eingegeben wurden.

4 Aufgabe - Backup-Skript

4.1

Erstellen Sie ein Shell-Skript, welches zwei Pfade übergeben bekommt. Der erste Pfad soll die Quelle sein und der zweite Pfad das Ziel. Wenn Skript aufgerufen wird, ohne die Übergabe von Ordnern, so soll die Hilfe für Ihr Skript ausgegeben werden. Ihr Skript soll dann eine Kopie von der Quelle beim Ziel anlegen, quasi als Backup. Ihr Skript soll alle Fehlermeldungen unter `/var/log/backup1.log` ablegen. Der Aufruf des Skripts soll dann wie folgt aussehen:

```
./backup.sh /home/azubi /opt/backup
```

Beispiel Ausgabe, bei erfolgreichem kopieren:

```
#####  
Ihr Ordner /home/azubi wurde nach /opt/backup/ gesichert.  
#####
```

4.2

Legen Sie eine Kopie Ihres zuvor erstellten Skriptes an und erweitern Sie Ihr Skript um die Funktion eines Auswahlmenüs (select), wo die Backups gespeichert werden sollen. Dies soll aufgerufen werden, wenn nur die Quelle angegeben wird. Die folgenden Auswahlmöglichkeiten sollen zur Verfügung stehen:

- `/opt/backup/`
- `/opt/sicherung/`
- `/tmp/backup`
- `/backups`

4.3

Legen Sie eine Kopie Ihres zuvor erstellten Skriptes an und erweitern Sie Ihr Skript um eine Komprimierungsfunktion.

Ihr Skript soll nun, statt eine Kopie, eine Archiv Datei erzeugen. Dies können Sie beispielsweise mit dem Befehl `tar` umsetzen.

4.4

Ab dieser Teilaufgabe, kann es häufiger zu Syntaxproblemen kommen, wenn Sie Ihren Quellcode zu viel anpassen. Überlegen Sie, bevor Sie Ihr Skript anpassen, was eine logische Herangehensweise an die folgende Problemstellung ist.

Legen Sie eine Kopie Ihres zuvor erstellten Skriptes an und erweitern Sie Ihr Skript um Optionen (getops). Die folgenden Optionen sollen zur Verfügung stehen:

- `-t` - Komprimierung in ein Tar-Archiv
- `-z` - Komprimierung in ein Zip-Archiv

- `-b` - Komprimierung in ein bz2-Archiv
- `-s` - Komprimierung in ein zstd-Archiv
- `-a` - Komprimierung in alle vorher genannten Archive (vier einzelne Dateien)
- `-h` - Hilfe anzeigen
- `-n` - Namen des Backups (ohne Endung)

4.5

Legen Sie eine Kopie Ihres zuvor erstellten Skriptes an und erweitern Sie Ihr Skript um einen Cron-Modus. Hierbei soll Ihr Skript, wenn es mit der Option `-c` aufgerufen wird, alle Ausgaben in das folgende Logfile schreiben `/var/log/azubi/bak.script.log`

Richten Sie zusätzlich einen Cron-Job ein, wo Ihr Skript täglich aufgerufen wird (morgens um 9 Uhr) und ein Backup von den folgenden Verzeichnissen anlegt:

- `/etc`
- `/home`
- `/root`

Das Backup soll unter `/opt/sicherungen/` abgelegt und als bz2-Archiv gespeichert werden.

5 Aufgabe - Passwortgenerator

5.1

Erstellen Sie ein Shell-Skript, welches ein Passwortgenerator enthält.
Dazu sollen Sie die folgenden Optionen implementieren:

- -l - für die Länge des Passwortes (inkl. Übergabe)
- -z - Zahlen sollen verwendet werden
- -s - Sonderzeichen sollen verwendet werden (# \$! - _ § &)
- -g - Großbuchstaben sollen verwendet werden
- -k - Kleinbuchstaben sollen verwendet werden
- -u - Umlaute sollen verwendet werden (äöü (wenn -k) und/oder ÄÖÜ (wenn -g))
- -a - Anzahl der Passwörter, die erstellt werden sollen (inkl. Übergabe)
- -p - Ausgabe ohne Optik
- -h - Hilfe soll angezeigt werden

Die Anzeige der Passwörter soll (ohne die Option -p) wie folgt aussehen:

```
#####  
# Passwort 1:   asdd                                     #  
# Passwort 2:   dfsf                                     #  
# ...                                                  #  
#####
```

Die Anzeige der Passwörter soll (mit der Option -p) wie folgt aussehen:

```
asdd  
dfs f
```

Ihr Skript soll auch Überprüfen, ob überhaupt eine Option angegeben ist.

5.2

Erweitern Sie Ihr Skript zum Benutzer anlegen (3.5) durch das automatische Erstellen von Passwörtern. Legen Sie eine Kopie Ihres Benutzer anlegen Skriptes an, bevor Sie das Skript erweitern.

Dazu müssen Sie Ihre user.1st um neue Benutzer erweitern. Immer dann, wenn kein Passwort angegeben ist, soll der Passwortgenerator genutzt werden zum Erstellen eines Passwortes.

6 Aufgabe - Logrotate

Konfigurieren Sie die Log-Rotation, für Ihr Backup-Skript (`/var/log/backup1.log`):

- Ausführung: täglich
- Logs die behalten werden sollen: 3
- Komprimieren der Datei
- Ignorieren des Fehlens der Logdatei
- Leere Logdateien werden rotiert
- Nach dem Rotieren soll Ihr Backup-Skript ausgeführt werden (wie beim cronjob).

7 Aufgabe - Netzwerktools

7.1

Worin unterscheiden sich `ping -R` und `traceroute`?

7.2

Worin unterscheiden sich die Befehle `dig`, `host` und `nslookup`?

7.3

Was machen die folgenden Befehle?

- `mtr`
- `iptraf`
- `ss`
- `getent hosts`

8 Aufgabe - ssh-Client

8.1

Konfigurieren Sie Ihren Client so, dass Sie den folgenden Befehl eintippen können und sich mit dem entfernten Server (172.20.XX.236) verbinden können.

ssh Server

Benutzername:

sshuser

Passwort:

ssh

8.2

Erzeugen Sie eine Datei mit der Ausgabe aller laufenden Prozesse `ps` auf Ihrem Schulungsrechner und übertragen Sie diese auf den Server (172.20.XX.236) in das Verzeichnis `/upload` mit dem Namen `<Vorname><Nachname>.nfo`

8.3

Worin unterscheiden sich die Programme `rsync`, `scp` und `sftp`?