



## M122 - Bash - Prüfung 2 Serie b

### Erlaubte Hilfsmittel :

- Alle Hilfsmittel ausser Netzwerkkommunikation (**LAN-Kabel entfernen!**)

### Generell nicht erlaubt ist:

- Abschreiben von der Nachbarprüfung oder Weitergeben von Prüfungen/Prüfungsergebnissen innerhalb der Klasse. Kommunikation über Handy oder Computernetzwerk.

Jeglicher Verstoß gegen diese Regeln führt umgehend zum Ausschluss aus der Prüfung und der Note 1.

Name: \_\_\_\_\_ Klasse: \_\_\_\_\_ Datum: \_\_\_\_\_

Aufgabe	max. Punkte	erreichte Punkte
1a	1	
1b	1	
1c	1	
1d	1	
1e	1	
2a	2	
2b	2	
2c	2	
2d	2	
3a	3	
3b	5	
Total	21	

## Teil 1 - Kurzaufgaben

**Aufgabe - 1a (1 Punkt)** grep - Gegeben ist folgendes Kommando:

```
cat /etc/passwd | grep -v ssh
```

Erklären sie, was dieses Kommando genau macht (berücksichtigen sie auch die Option `-v`):

Antwort: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**Aufgabe - 1b (1 Punkt, 1/2 Punkt pro Pfadangabe)** - Verzeichnisse wechseln  
Angenommen, sie befinden sich auf Ihre Linuxsystem unter `/home/user1/`, geben sie die **relative** und die **absolute** Pfadangabe an, um in das Verzeichnis `/tmp` zu wechseln:

Antwort: \_\_\_\_\_  
\_\_\_\_\_

**Aufgabe - 1c (1 Punkt)** Cronjob - Sie wollen eines Ihre Bashskripte mit crontab automatisieren, dazu haben sie einen cronjob eingerichtet welcher nach nachfolgendem Muster ausgelöst wird:

```
0 2 * * 3
```

Wann wird das Skript jeweils ausgeführt?

Antwort: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**Aufgabe - 1d (1 Punkt pro richtige, 1 Punkt pro falsche oder fehlende richtige Antwort)**  
- Welches Dateirecht muss zusätzlich auf einer Skriptdatei gesetzt werden, damit diese ausgeführt werden kann?

- 0
- x
- w
- t
- s

**Aufgabe - 1e (1 Punkt)** Wildcards - Sie führen folgendes Kommando aus:

```
touch {d..f}file
```

Erklären sie, was das Kommando genau macht:

Antwort: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

## Teil 2 - Codestücke

### Aufgabe - 2a (2 Punkte) - Backup Skript

Nachfolgend sehen sie ein unvollständiges Backupskript. Die wichtigsten Informationen, um mit tar ein Backup durchzuführen, werden in den Variablen `quelle`, `ziel` und `outputFile` zwischengespeichert. Vervollständigen sie das Skript so, dass ein Backup vom Ordner `home` gemacht wird und zB. in der Datei `/var/backups/home-20151214.tgz` abgelegt wird. Benutzen sie dazu die bereits definierten Variablen.

```
#!/bin/bash
outputFile=home-$(date +%Y%m%d).tgz
quelle="/home/"
ziel="/var/backups/"
tar -cZf _____
```

### Aufgabe - 2b (2 Punkte) - for in

Nachfolgend sehen sie ein unvollständiges Skript. Sie sollten das Skript so ergänzen, dass es den Inhalt des Verzeichnisses `/home/` ausgibt und zwar unter der Verwendung der abgebildeten `for in` Schleife.

```
1  #!/bin/bash
2  for i in _____ ; do
3      echo $i
4  done
```

### Aufgabe - 2c (2 Punkte, 1 Punkt pro Zeile) - If - else

Nachfolgend sehen sie ein unvollständiges Skript. Sie sollen das Skript nun so ergänzen (Zeilen 3 und 5), dass dieses, falls das 1. Argument den Wert `keywOrd` hat, den Text *sie dürfen passieren* ausgibt und falls nicht den Text *nicht erlaubt!*.

```
1  #!/bin/bash
2  if [ $1 = "keywOrd" ]; then
3      _____
4  else
5      _____
6  fi
```

### Aufgabe - 2d (2 Punkte, 1 Punkt pro Zeile) - while

Nachfolgend sehen sie ein unvollständiges Skript. Sie sollen das Skript so ergänzen, dass die Schleife solange läuft, wie `zaehler` kleiner als 10 ist (Zeile 3) und der `zaehler` zum Schluss jedes Schleifendurchlaufs um 1 erhöht wird (Zeile 5).

```
1  #!/bin/bash
2  zaehler=0
3  while [ $zaehler _____ ]; do
4      echo der zaehler ist $zaehler
5      _____
6  done
```

## Teil 3 - Programmieraufgaben

### Aufgabe - 3a (3 Punkte, 1 Punkt pro Zeile) - Rechnerskript

Im nachfolgenden sehen sie ein Taschenrechnerskript, welches zwei Zahlen und eine Operation einliest und dann die beiden Zahlen mit dieser Operation verrechnet (zB. addiert) und ausgibt:

```
1  echo "bitte gebe eine Nummer ein"
2  read n1
3  echo "bitte wähle eine Operation"
4  echo "1. addieren"
5  echo "2. subtrahieren"
6  echo "3. dividieren"
7  echo "4. multiplizieren"
8  read opr
9  echo "bitte gebe eine zweite Nummer ein"
10 read n2
11
12 if [ $opr = "1" ]
13     then
14         echo $((n1+n2))
15 -----
16         -----
17         -----
18 elif [ $opr = "3" ]
19     then
20         echo $((n1/n2))
21 elif [ $opr = "4" ]
22     then
23         echo $((n1*n2))
24 fi
```

Der Code einer Operation fehlt (Subtraktion). Ergänzen sie den Code an der hervorgehobenen Stelle so, dass auch die Subtraktion funktioniert.

### Aufgabe - 3b (5 Punkte) - IP-Adressen von Nameservern

In der Datei `/etc/resolv.conf` sind die IPs der von Ihrem Linuxsystem verwendeten Nameserver eingetragen. Der Inhalt dieser Datei kann zB. so aussehen:

```
search example.com
nameserver 10.0.1.6
nameserver 10.0.1.7
```

Schreiben sie ein kleines Skript, welches alle IP's der Nameserver ausgibt. Mit dem obigen Beispiel sollte Ihr Skript also folgende Ausgabe erzeugen:

```
10.0.1.6
10.0.1.7
```

Verwenden sie dazu `cat`, `grep` und `cut`.