

M122 - Bash - Prüfung 2 Serie a

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	5	
3b	3	
Total	21	

Teil 1 - Kurzaufgaben

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

```
0 4 * * 1
```

Wann wird das Skript jeweils ausgeführt?

Antwort: _____

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

```
touch {a..c}file
```

Erklären sie, was das Kommando genau macht:

Antwort: _____

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

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

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

Antwort: _____

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

Antwort: _____

Aufgabe - 1e (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



Teil 2 - Codestücke

Aufgabe - 2a (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 `passw0rd` hat, den Text *Passwort korrekt*, *Zutritt erlaubt* ausgibt und falls nicht den Text *Zutritt verweigert!*.

```
1  #!/bin/bash
2  if [ $1 = "passw0rd" ]; then
3      -----
4  else
5      -----
6  fi
```

Aufgabe - 2b (2 Punkte) - for in

Nachfolgend sehen sie ein unvollständiges Skript. Sie sollten das Skript so ergänzen, dass es den Inhalt des Verzeichnisses `/etc/` 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) - while

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

```
1  #!/bin/bash
2  zaehler=20
3  while [ $zaehler _____ ]; do
4      echo der zaehler ist $zaehler
5      -----
6  done
```

Aufgabe - 2d (2 Punkte) - Backup Skript

Nachfolgend sehen sie ein unvollständiges Backupskript. Die wichtigsten Informationen, um mit `tar` ein Backup durchzuführen, werden in den Variablen `source`, `target` 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
source="/home/"
target="/var/backups/"
outputFile=home-$(date +%Y%m%d).tgz
tar -cZf _____
```

Teil 3 - Programmieraufgaben

Aufgabe - 3a (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 10.0.3.4
nameserver 10.0.3.5
```

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

```
10.0.3.4
10.0.3.5
```

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

Aufgabe - 3b (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 "Geben sie die erste Nummer ein:"
2  read a
3  echo "Geben sie die zweite Nummer ein:"
4  read b
5  echo "Geben sie die gewünschte Operation +, -, / oder x ein"
6      read opr
7  if [ $opr = "+" ]
8      then
9      op='expr $a + $b'
10     echo "$op"
11  -----
12     -----
13     -----
14     -----
15  elif [ $opr = "/" ]
16      then
17      op='expr $a / $b'
18     echo "$op"
19  elif [ $opr = "x" ]
20      then
21      op='expr $a \* $b'
22     echo "$op"
23  fi
```

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