

Probeklausur

03.03.2025/10.03.2025

Hinweise

- Es sind keine Hilfsmittel erlaubt!
- es gibt 30 Punkte

Aufgabe 1 Basics

Nenne und begründe einen Vor- und Nachteil von Skriptsprachen gegenüber kompilierten Sprachen.
(1 Punkt)

Aufgabe 2 Basics

Was bewirkt der Befehl `cd` wenn er ohne Argumente ausgeführt wird? (1 Punkt)

Aufgabe 3 Basics

Ist die folgende Aussage wahr?

„Eine Datei, die ausgeführt werden soll, muss immer eine Dateiendung besitzen (z.B.: `.sh`).“ (1 Punkt)

Aufgabe 4 Basics

Welche Aussagen müssen zutreffen, um das Skript `script.sh` mit `bash script.sh` auszuführen.

1. `script.sh` der/die UserIn muss das Ausführungsrecht für `script.sh` haben (0,5 Punkt)
2. `script.sh` der/die UserIn muss das Leserecht für `script.sh` haben (0,5 Punkt)
3. `script.sh` muss eine Shebang-Zeile haben (0,5 Punkt)
4. `script.sh` muss in einem Ordner liegen, der sich in der `PATH` Variablen befindet (0,5 Punkt)

Aufgabe 5 Shebang

Es liegt folgendes Skript (Dateiname: `script.sh`) vor:

```
#!/bin/sh

echo "hello"
```

Mit welchem Programm/Interpreter wird das Skript ausgeführt, wenn es folgendermaßen gestartet wird:

```
bash ./script.sh
```

(1 Punkt)

Aufgabe 6 Spezielle Variablen und Maskierung

Was machen die beiden Codezeilen und wie unterscheiden sie sich?

A) B)

```
bash -c 'echo $$'          echo $$
```

„`bash -c`“ führt den übergebenen Parameter als Befehl aus. Der Parameter wird **nicht** wie gewohnt als Dateiname interpretiert.

(2 Punkt)

Aufgabe 7 Spezielle Variablen und Funktionen

Wie ist die Ausgabe des folgenden Scripts?

```
foo(){
    echo bar
    return 1
}

FOO=$(foo)
echo $?
echo $FOO
```

(1 Punkt)

Aufgabe 8 Spezielle Variablen

Was ist der Exit-Status eines erfolgreich ausgeführten Befehls und wie kann dieser ausgegeben werden?

(1 Punkt)

Aufgabe 9 Globbing

Wie können alle Dateien, deren Namen mit `.pdf` enden und sich im aktuellen Verzeichnis befinden mit `ls` aufgelistet werden? (1 Punkt)

Aufgabe 10

Was ist der Unterschied zwischen den zwei Befehlen

1. `touch datei`
2. `echo > datei`

(1 Punkt)

Aufgabe 11 Streams

Was ist der Unterschied zwischen `echo foo > file.txt` und `echo foo >> file.txt`? (1 Punkt)

Aufgabe 12 Streams

In welche Dateien wird `stderr` und `stdout` umgeleitet?

```
./mysterious_script.sh 1>> a 2>> b 3>> c
```

(1 Punkt)

Aufgabe 13 Streams

Warum ist der Befehl `“echo foo > output.txt”` gleichbedeutend zu `“echo foo 1> output.txt”`? (1 Punkt)

Was passiert hier: `“befehl > output.log 2>&1”`? (Nenne auch die Datenströme) (1 Punkt)

Was passiert hier: `“befehl1 | befehl2”`? (Nenne auch die Datenströme) (1 Punkt)

Aufgabe 14 Stringmanipulierung

1. Wie ist die Ausgabe von dem folgenden Skript.

schneidet den größtmöglichen Teil eins Strings vom Anfang ab.

```
STR=/a/b/c/file.tex
echo ${STR##*/}
```

(1 Punkt)

2. Wie ist die Ausgabe von dem folgenden Skript.

schneidet den kleinstmöglichen Teil eins Strings vom Anfang ab.

```
STR=/a/b/c/file.tex
echo ${STR#*/}
```

(1 Punkt)

Aufgabe 15 Signale

Ein Skript fängt das Signal 2 (SIGINT) mit

```
trap 'echo "cannott exit"' 2
```

ab.

Auf welche Art kann das Skript trotzdem vorzeitig beendet werden? Nenne nicht nur den Befehl oder die Tastenkombination, sondern auch was sie bewirkt und woher du die Argumente für die verwendeten Befehle beziehst. (2 Punkte)

Aufgabe 16 Friendly Shell

Das folgende Skript soll eine interaktive Shell sein. Alle Befehle müssen mit „please“ beginnen. Andererseits soll ein zufällig ausgewählter String ausgegeben werden. Das Signal 2 soll auch nicht wie gewohnt zum beenden des Skripts führen. Die Ausgaben des Skripts beschränken sich auf die „zufälligen Strings“ und die Rückgaben der ausgeführten Befehlen.

In dem Skript sind 3 Fehler (Stellen im Code die einen Widerspruch mit der oberen Beschreibung erzeugen) . Finde die Fehler und markiere sie. Beschreibe zu welchem Verhalten die Fehler führen und wie sie korrigierbar sind. (3 Punkte)

```
#!/bin/bash

replies='no
no way
never'

reply() {
    echo "$replies" | head -n 1
}

trap reply 2

while true; do
    read -r -p "hi " input
    echo $input | grep "please" || { echo $(reply); continue; }
    ${input#please}
done
```

Aufgabe 17

Warum muss in der Regel der Pfad angegeben werden, um ein Skript auszuführen? Was muss getan werden, damit es reicht nur den Namen, anstatt den Pfad, eines Skripts anzugeben, um es auszuführen? (1 Punkt)

Aufgabe 18

Wann wird die Datei `.bashrc` gelesen? (1 Punkt)

Aufgabe 19

Wie kann ein Script ausgeführt werden, ohne eine neue Shell-Instanz zu starten? Wann ist es sinnvoll auf diese Weise ein Skript auszuführen? (2 Punkte)

Aufgabe 20 find und xargs

Beschreibe was der folgende Befehl tut

```
find . -type f -name "*.sh" | xargs cp -t ~/scripts
```

Wie oft wird `cp` aufgerufen, wenn `find` zehn Dateien findet? (2 Punkte)