

Landesberufsschule 4 Salzburg

Übungen im IT-Laboratorium

Bash-Skripte

für die Übung Nr. 12

Katalog - Nr.: 1

Name : Valentin Adlgasser

Jahrgang : 2018/19

Datum der Übung : 20.12.2018

Inhalt

1. Anweisung der Übung:	2
2. Einleitung.....	2
3. Inventarliste.....	2
4. Übungsdurchführung	3
Unterpunkt	Fehler! Textmarke nicht definiert.
5. Einsatzgebiet	4
6. Erkenntnisse	4

1. Anweisung der Übung:

Siehe Moodle

2. Einleitung

In diesem Protokoll kann man nachlesen wie man Bash-Skripte unter Ubuntu-Server schreibt. Wie man diese ausführt und wofür diese verwendet werden.

Skripte benötigen, im Gegensatz zu Programmen, keinen Compiler. Stattdessen wird die Textdatei Zeile für Zeile eingelesen, interpretiert und ausgeführt.

3. Inventarliste

VM (Ubuntu-Server)

4. Übungsdurchführung

a. Skript erstellen und ausführbar machen

Um ein Skript unter Ubuntu-Server zu erstellen benutzt man am besten den Editor „Nano“. In jedem Skript muss als erste Zeile der Eintrag „#!/bin/bash“ stehen. Dieser gibt an, an welchem Speicherort die Bash installiert ist.

Nachdem das Skript fertiggestellt und gespeichert worden ist, muss man es noch ausführbar machen. Dafür gibt man im Terminal den Befehl „chmod 755 [Skriptname]“ ein.

b. 1. Skript

Das erste Skript gibt einfach im Terminal, den angemeldeten Benutzer, das jetzige Arbeitsverzeichnis und das aktuelle Datum + Uhrzeit aus.

```
#!/bin/bash
printf "User:      "
who
printf "Directory:  "
pwd
printf "Date:       "
date
```

Abbildung 1 | Skript 1

c. 2. Skript

Das zweite Skript nimmt Parameter vom User entgegen und gibt danach die Anzahl der Parameter und die Parameter an sich aus.

```
#!/bin/bash

((counter++))

for i in $*
do
    echo "Parameter$counter: $i"
    ((counter++))
done

echo "Anzahl Parameter: $#"
```

Abbildung 2 | Skript 2

d. 3. Skript

Das dritte Skript stellt dem User zwei Fragen und nimmt jeweils eine Antwort entgegen. Danach gibt es die Antworten des Users, in einem Satz, in Farbe aus.

```
#!/bin/bash

read -p "Geben Sie Ihren Namen ein: " name
echo -e "\e[32mHallo $name\e[39m"
read -p "Wie geht es dir $name?" zustand
echo -e "\e[36m$name geht es $zustand\e[39m"
```

Abbildung 3 | Skript 3

e. 4. Skript

Das vierte Skript nimmt einen Parameter entgegen, wenn dieser „J“ oder „j“ ist gibt es den Inhalt des aktuellen Verzeichnisses aus. Wird ein anderer Parameter eingegeben, beendet sich das Skript und gibt aus, welcher Parameter mitgegeben wurde.

Wird allerdings gar kein Parameter mitgegeben, dann fragt das Skript ob der Inhalt des Verzeichnisses ausgegeben werden soll. Hier kann der User wieder mit „J“ oder „j“ für Ja antworten, oder er gibt etwas Anderes ein und das Skript wird beendet.

```
#!/bin/bash
if [ "$#" -ne "1" ]
then
    read -p "Soll der Inhalt des Verzeichnisses ausgegeben werden? (J/N)" yn
    if [ "$yn" == "J" -o "$yn" == "j" ]
    then
        ls -ll
    else
        echo -e "Programm wird beendet \e[91mEingabe war: $yn\e[39m"
    fi
else
    if [ "$1" == "J" -o "$1" == "j" ]
    then
        ls -ll
    else
        echo -e "Programm wird beendet \e[91mEingabe war: $1\e[39m"
    fi
fi
```

Abbildung 4 | Skript 4

5. Einsatzgebiet

Skripte können vielseitig verwendet werden. Z.B. für die Automatisierung von Programmen oder anderen Dingen in der Firma, oder um kleine „Programme“ zu schreiben welche nicht viel Speicherplatz oder Laufzeit benötigen.

6. Erkenntnisse

Skripte zu schreiben ist relativ intuitiv, wenn man schon einmal grundsätzlich programmiert hat.

Unterschrift: