Titel: Labor 8

Klasse: 3BHIF

Name: Haiden

Gruppe: 01

Aufgabe: 19.02.2020 Abgabe: 04.03.2020

Inhaltsverzeichnis

[1 Shell-Skripte – Labor 8 1](#_Toc34162592)

[1.1 Aufgabe 1 1](#_Toc34162593)

[1.1.1 Skript 1](#_Toc34162594)

[1.1.2 Testläufe 2](#_Toc34162595)

[1.2 Aufgabe 2 3](#_Toc34162596)

[1.2.1 Skript 3](#_Toc34162597)

[1.2.2 Testläufe 4](#_Toc34162598)

[1.3 Aufgabe 3 5](#_Toc34162599)

[1.3.1 Skript: 5](#_Toc34162600)

[1.3.2 Testläufe 7](#_Toc34162601)

[1.4 Aufgabe 4 8](#_Toc34162602)

[1.4.1 Skript 8](#_Toc34162603)

[1.4.2 Testläufe 9](#_Toc34162604)

[1.5 Aufgabe 5 9](#_Toc34162605)

[1.5.1 Skript: 9](#_Toc34162606)

[1.5.2 Testläufe 10](#_Toc34162607)

[1.6 Aufgabe 6 11](#_Toc34162608)

[1.6.1 Skript 11](#_Toc34162609)

[1.6.2 Testläufe 12](#_Toc34162610)

[1.7 Aufgabe 7 12](#_Toc34162611)

[1.7.1 Skript: 12](#_Toc34162612)

[1.7.2 Testläufe 13](#_Toc34162613)

# Shell-Skripte – Labor 8

## Aufgabe 1

Schreiben Sie ein Shellskript, das

* alle übergebenen Argumente
* die Anzahl der Argumente
* den Namen des Shellskripts
* den Inhalt des Shellskripts

auf stdout ausgibt.

### Skript

#!/bin/bash

#

#Uebung 08 Beispiel 1

#Haiden Niklas 3BHIF-06

#2020-02-19

if test $# -eq 0

then

echo "Es wurden keine Argumente angegeben."

exit 1

fi

anz\_Args="Anzahl der Argumente: "$#

uebergebene\_Args="Uebergebene Argumente "$\*

skript\_Name="Name des Skripts: "$0

echo $anz\_Args

echo $uebergebene\_Args

echo $skript\_Name

cat $0

exit 0

### Testläufe

#### Aufruf mit Parametern

schueler@Debian10nvs:~/skripte\_uebung8$ ./aufgabe1\_labor8.sh hello welt

Anzahl der Argumente: 2

Uebergebene Argumente hello welt

Name des Skripts: ./aufgabe1\_labor8.sh

#!/bin/bash

#

#Uebung 08 Beispiel 1

#Haiden Niklas 3BHIF-06

#2020-02-19

if test $# -eq 0

then

echo "Es wurden keine Argumente angegeben."

exit 1

fi

anz\_Args="Anzahl der Argumente: "$#

uebergebene\_Args="Uebergebene Argumente "$\*

skript\_Name="Name des Skripts: "$0

echo $anz\_Args

echo $uebergebene\_Args

echo $skript\_Name

cat $0

exit 0

#### Aufruf ohne Parameter

schueler@Debian10nvs:~/skripte\_uebung8$ ./aufgabe1\_labor8.sh

Es wurden keine Argumente angegeben.

## Aufgabe 2

Das Shellskript soll genau ein Argument entgegennehmen und dieses dann zusammen mit einer Ausgabe in der Form

Es wurde der Parameter "hallo" eingegeben

ausgeben. Zusätzlich soll jedes Mal, wenn das Script ausgeführt wird, ein Logeintrag mit dem Usernamen, dem aktuellem Timestamp im Format yyyy-MM-dd hh:mm:ss und dem eingegebenen Parameter an die Datei ue\_02.log angehängt werden.Wurde nicht genau ein Parameter angegeben, so beinhaltet der Logeintrag den Usernamen, den Timestamp, eine Fehlermeldung und die Anzahl der übergebenen Argumente.

### Skript

#!/bin/bash

#

#Uebung 08 Beispiel 2

#Haiden Niklas 3BHIF-06

#2020-02-19

ausgabe\_shell="Es wurde der Parameter \""$1"\" ausgegeben"

username=$USER

timestamp=$(date +"%Y-%m-%d %H-%M-%S")

fehlermeldung="nix"

ausgabe\_file=$USER", "$timestamp", "$1

if test $# -ne 1

then

if test $# -eq 0

then

fehlermeldung="Es wurde kein Parameter ausgegeben."

fi

if test $# -gt 1

then

fehlermeldung="Es wurde mehr als ein Parameter ausgegeben."

Fi

echo $fehlermeldung

ausgabe\_file=$USER", "$timestamp", "$fehlermeldung", "$#

echo $ausgabe\_file>>ue\_02.log

exit 1

fi

echo $ausgabe\_file >> ue\_02.log

echo $ausgabe\_shell

### Testläufe

#### Aufruf ohne Parameter

schueler@Debian10nvs:~/skripte\_uebung8$ ./aufgabe2\_labor8.sh

Es wurde kein Parameter ausgegeben.

schueler@Debian10nvs:~/skripte\_uebung8$ cat ue\_02.log

schueler, 2020-02-19 09-46-21, Es wurde kein Parameter ausgegeben., 0

#### Aufruf mit Parameter:

schueler@Debian10nvs:~/skripte\_uebung8$ ./aufgabe2\_labor8.sh hello

Es wurde der Parameter "hello" ausgegeben

schueler@Debian10nvs:~/skripte\_uebung8$ cat ue\_02.log

schueler, 2020-02-19 09-46-21, Es wurde kein Parameter ausgegeben., 0

schueler, 2020-02-19 09-50-04, hello

#### Aufruf mit mehr als zwei Parametern

schueler@Debian10nvs:~/skripte\_uebung8$ ./aufgabe2\_labor8.sh hello world

Es wurde mehr als ein Parameter ausgegeben.

schueler@Debian10nvs:~/skripte\_uebung8$ cat ue\_02.log

2020-02-19 09-31-06

schueler, 2020-02-19 09-46-21, Es wurde kein Parameter ausgegeben., 0

schueler, 2020-02-19 09-50-04, hello

schueler, 2020-02-19 09-52-15, Es wurde mehr als ein Parameter ausgegeben., 2

## Aufgabe 3

Schreiben sie ein Script, dasauf 3 unterschiedliche Argumente reagiert:

* Wenn es mit dem Argument login aufgerufen wird, dann soll es jene Zeile aus der Datei /etc/passwd ausgeben, die dem gerade aktivem User zugeordnet ist.
* Wenn es mit dem Argument ps-root aufgerufen wird, soll dasKommando ps -efaufgerufen werden, und aus dessen Ausgabe sollen alle Zeilen, die das Wort root enthalten, herausgefiltert und dann seitenweise angezeigt werden.
* Wenn es mit dem Argument -h aufgerufen wird, dann soll ein kurzer Hilfetext zur Benutzung erscheinen.
* Wenn es ohne Argumente aufgerufen wird, dann soll ebenfalls der Hilfetext ausgegeben werden.

Erklären sie in einem Kommentar auch die Bedeutung des Kommandos ps –ef.

### Skript:

#!/bin/bash

#

#Uebung 08 Beispiel 3

#Haiden Niklas 3BHIF-06

#2020-02-19

# Funktion, um die Hilfe nicht zweimal schreiben zu müssen

hilfe\_ausgabe () {

echo "Dies ist die Hilfe fuer Aufgabe 3

--------------------------------------------------------------------------------------------------------------

Mit dem Parameter -login wird die Passwd-Zeile des aktuellen Users ausgegeben.

---------------------------------------------------------------------------------------------------------------

Mit dem Parameter -h wird dieser Hilfetext ausgegeben.

---------------------------------------------------------------------------------------------------------------

Mit dem Parameter -ps-root werden alle Zeilen vom kommando ps -ef herausgefiltert, die das Wort root enthalten."

}

if test $# -gt 1

then

echo "Mehr als 1 Argument wurde angegeben."

exit 1

fi

if test $# -eq 0

then

hilfe\_ausgabe

exit 0

fi

if test $1 = "-h"

then

hilfe\_ausgabe

exit 0

fi

if test $1 = "-login"

then

cat /etc/passwd | grep $USER

exit 0

fi

if test $1 = "-ps-root"

then

ps -ef | grep "^root"

exit 0

fi

exit 0

### Testläufe

#### Aufruf ohne Parameter

schueler@Debian10nvs:~/skripte\_uebung8$ ./aufgabe3\_labor8.sh

Dies ist die Hilfe fuer Aufgabe 3

--------------------------------------------------------------------------------------------------------------

Mit dem Parameter -login wird die Passwd-Zeile des aktuellen Users ausgegeben.

---------------------------------------------------------------------------------------------------------------

Mit dem Parameter -h wird dieser Hilfetext ausgegeben.

---------------------------------------------------------------------------------------------------------------

Mit dem Parameter -ps-root werden alle Zeilen vom kommando ps -ef herausgefiltert, die das Wort root enthalten.

#### Aufruf mit Parameter –h

schueler@Debian10nvs:~/skripte\_uebung8$ ./aufgabe3\_labor8.sh -h

Dies ist die Hilfe fuer Aufgabe 3

--------------------------------------------------------------------------------------------------------------

Mit dem Parameter -login wird die Passwd-Zeile des aktuellen Users ausgegeben.

---------------------------------------------------------------------------------------------------------------

Mit dem Parameter -h wird dieser Hilfetext ausgegeben.

---------------------------------------------------------------------------------------------------------------

Mit dem Parameter -ps-root werden alle Zeilen vom kommando ps -ef herausgefiltert, die das Wort root enthalten.

#### Testlauf mit –login

schueler@Debian10nvs:~/skripte\_uebung8$ ./aufgabe3\_labor8.sh -login

schueler:x:1000:1000:Schueler der Abteilung Informatik,,,:/home/schueler:/bin/bash

#### Testlauf mit –ps-root

schueler@Debian10nvs:~/skripte\_uebung8$ ./aufgabe3\_labor8.sh -ps-root

root 1 0 0 07:57 ? 00:00:03 /sbin/init

root 2 0 0 07:57 ? 00:00:00 [kthreadd]

## Aufgabe 4

Schreiben sie ein Script, das je nach aktueller Systemzeit "Good morning" (ab 00:00), "Good afternoon" (ab 12:00) oder "Good evening" (ab 18:00) ausgibt.

### Skript

#!/bin/bash

#

#Uebung 08 Beispiel 4

#Haiden Niklas 3BHIF-06

#2020-02-19

time=$(date +"%H")

if test $time -ge 08

then

echo "Good morning"

fi

if test $time -ge 12

then

echo "Good afternoon"

fi

if test $time -ge 18

then

echo "Good evening"

fi

### Testläufe

#### Aufruf: ./aufgabe4\_labor8.sh

schueler@Debian10nvs:~/skripte\_uebung8$ ./aufgabe4\_labor8.sh

Good morning

## Aufgabe 5

Schreiben Sie ein Shellskript, das in der als 1. Parameter übergebenen Datei nach dem Wort sucht, das als 2.Parameter übergeben wurde. Das Skript soll alle Zeilen und deren Anzahl ausgeben, in denen das gesuchte Wort vorkommt.

Returnwerte:

* 0 fehlerfrei
* 1 falsche Anzahl an Parametern
* Datei $1 existiert nicht
* Datei $1 existiert, darf aber nicht gelesen werden

### Skript:

#!/bin/bash

#

#Uebung 08 Beispiel 5

#Haiden Niklas 3BHIF-06

#2020-02-19

FILE=$1

SEARCH\_TERM=$2

if [ $# -ne 2 ]

then

echo "Es wurden nicht 2 Parameter angegeben."

exit 1

fi

if [ ! -f $FILE ]

then

echo "Die Datei existiert nicht."

exit 2

fi

if [ ! -r $FILE ]

then

echo "Die Datei ist existiert nicht und / oder ist nicht lesbar."

exit 3

fi

grep $SEARCH\_TERM $FILE

anzahl=$(grep $SEARCH\_TERM $FILE | wc -l)

echo "Anzahl: " $anzahl

### Testläufe

#### Normaler Fall – Datei existiert

schueler@Debian10nvs:~/skripte\_uebung8$ ./aufgabe5\_labor8.sh testfile.txt Katze

Hallo hier sollte Katze stehen

Hier steht wieder Katze

Anzahl: 2

schueler@Debian10nvs:~/skripte\_uebung8$ echo $?

0

#### Datei existiert nicht

schueler@Debian10nvs:~/skripte\_uebung8$ ./aufgabe5\_labor8.sh somefile.txt Katze

Die Datei existiert nicht.

schueler@Debian10nvs:~/skripte\_uebung8$ echo $?

2

#### Datei ist nicht lesbar

schueler@Debian10nvs:~/skripte\_uebung8$ chmod 000 testfile.txt

schueler@Debian10nvs:~/skripte\_uebung8$ ./aufgabe5\_labor8.sh somefile.txt Katze

Die Datei existiert nicht.

schueler@Debian10nvs:~/skripte\_uebung8$ ./aufgabe5\_labor8.sh testfile.txt Katze

Die Datei ist existiert nicht und / oder ist nicht lesbar.

schueler@Debian10nvs:~/skripte\_uebung8$ echo $?

3

#### Anzahl der Parameter ist nicht richtig

schueler@Debian10nvs:~/skripte\_uebung8$ ./aufgabe5\_labor8.sh

Es wurden nicht 2 Parameter angegeben.

schueler@Debian10nvs:~/skripte\_uebung8$ echo $?

1

## Aufgabe 6

Schreiben Sie ein einfaches Shellskript, das eine beliebige Anzahl an Argumenten übernimmt und das jedes Argument mit einem vorangestellten Hallo in einer eigenen Zeile ausgibt. Das Skript soll z.B. wie folgt arbeiten:

./ue\_03\_1\_3a00.sh Hugo Franz

Hallo Hugo

Hallo Franz

### Skript

#!/bin/bash

#

#Uebung 08 Beispiel 6

#Haiden Niklas 3BHIF-06

#2020-03-03

if [ $# -lt 1 ]

then

echo "Es muss mindestens ein Parameter vorhanden sein."

exit 1

fi

str=$\*

IFS=' '

read -ra ADDR <<< "$str"

for i in "${ADDR[@]}"; do

echo "Hallo, "$i

done

Dieses Skript benutzt den eingebauten read Command der Bash. Über die Variable IFS wird ein Delimiter festgelegt, welcher bei dem Splitten benutzt werden soll. Mit der Option -a sagt man dem eingebauten Command, dass er die Wörter in ein Array in diesem Fall ADDR schreiben soll. In der For-Schleife werden die einzelnen Elemente aus dem Array herausgeholt und in die Variable i gespeichert. Diese wird dann mittels echo Befehl ausgegeben.

### Testläufe

#### Guter Testlauf

niklas@nhaiden:~/nvs/labor8$ ./aufgabe6\_labor8.sh Susi Franzl Bernd

Hallo, Susi

Hallo, Franzl

Hallo, Bernd

#### Testlauf mit zu wenig Parametern

niklas@nhaiden:~/nvs/labor8$ ./aufgabe6\_labor8.sh

Es muss mindestens ein Parameter vorhanden sein.

## Aufgabe 7

Schreiben Sie ein Shellskript, das genau zwei ganze Zahlen a und b als Parameter übernimmt und das mit Hilfe einer while-Schleife alle Zahlen von a bis b (inklusive), jeweils genau durch ein Blank getrennt ausgibt. Sollten nicht genau 2 Parameter übergeben worden sein, beendet sich das Skript mit 1, wenn einer der beiden Parameter keine ganze Zahl ist mit 2 und wenn a > b ist mit 3.

Beispiel: ./ue\_03\_2\_3a00.sh 2 8

2345678

### Skript:

#!/bin/bash

#

#Uebung 08 Beispiel 7

#Haiden Niklas 3BHIF-06

#2020-03-03

if [ $# -ne 2 ]

then

echo "Zu wenige / zu viele Parameter."

exit 1

fi

if ! [[ "$1" =~ ^[+-]?[0-9]+$ ]] || ! [[ "$2" =~ ^[+-]?[0-9]+$ ]];

then

echo "Es sind nur Integer erlaubt!"

exit 2

fi

if [ $1 -gt $2 ]

then

echo "Der erste Parameter darf nicht groesser als der zweite sein!"

exit 3

fi

counter=$1

while [ $counter -le $2 ]

do

echo $counter

counter=$(($counter+1))

done

### Testläufe

#### Normaler Testlauf mit richtigen Parametern

niklas@nhaiden:~/nvs/labor8$ ./aufgabe7\_labor8.sh 2 8

2

3

4

5

6

7

8

#### Testlauf mit falscher Anzahl an Parametern

niklas@nhaiden:~/nvs/labor8$ ./aufgabe7\_labor8.sh 2

Zu wenige / zu viele Parameter.

niklas@nhaiden:~/nvs/labor8$ ./aufgabe7\_labor8.sh

Zu wenige / zu viele Parameter.

#### Testlauf mit nicht gueltigen Parametern

niklas@nhaiden:~/nvs/labor8$ ./aufgabe7\_labor8.sh xx 2

Es sind nur Integer erlaubt!

niklas@nhaiden:~/nvs/labor8$ ./aufgabe7\_labor8.sh 2 xx

Es sind nur Integer erlaubt!

#### Testlauf wo a größer b

niklas@nhaiden:~/nvs/labor8$ ./aufgabe7\_labor8.sh 3 1

Der erste Parameter darf nicht groesser als der zweite sein!