

Übungsblatt 01

Übungsblatt 02

Aufgabe 4: alias uhrzeit="date"

Übungsblatt 03

Aufgabe 2: mkdir -p ~/verz/uv1 ~/verz/uv2 ~/verz/uv3
Aufgabe 5: ls -R
Aufgabe 8: find . -name "datei1*"
Aufgabe 9: rm -r ~/verz

Übungsblatt 04

Aufgabe 1: split -a 1 -l 10 eingabedatei teil.
Aufgabe 2: cat teil.a teil.b > ausgabedatei
Aufgabe 3: cp * splitdateien/ ; rm *
Aufgabe 4: mv * splitdateien/
Aufgabe 5: rm -r splitdateien

Übungsblatt 05

Aufgabe 1: mkdir -p ~/uebung5/uv1 ~/uebung5/uv2
Aufgabe 2: cp /bin/[ac]* ~/uebung5/
Aufgabe 3: cp /bin/?* ~/uebung5/uv1/
Aufgabe 4: cp /bin/[!ds]* ~/uebung5/uv2/
Aufgabe 5: ls -ali /bin > ~/uebung5/text
Aufgabe 6: find ~/uebung5 -type f -name "v*"
find ~/uebung5 -type f -name "??*" -size -50k
find ~/uebung5 -type f -name "[abcde]?"
find ~/uebung5 -type f -size +30k -size -50k
Aufgabe 7: find / -type d -name "v*"
Aufgabe 8: file /etc/* | less

Übungsblatt 06

Aufgabe 3: find /etc/ -type f -exec grep -l "Have a lot of fun..." {} \;
Aufgabe 4: ls -laR / | grep "suchname"
Aufgabe 5: file /etc/* | grep -i ASCII | sort > ~/tmp/asciilist
Aufgabe 8: chmod u+x ~/tmp/script1 oder z.B. chmod 0700 ~/tmp/script1

Übungsblatt 07

Aufgabe 2: cat /etc/passwd | cut -d : -f 1,6 | sort
Aufgabe 3: cat /etc/passwd | cut -d : -f 1,7 | sed 's:/ hat die Login-Shell /g'
Aufgabe 5: df -h
Aufgabe 6: du -sh /bin
Aufgabe 7: tar -czf ~/tmp/bin.tar.gz /bin
Aufgabe 8: cd tmp ; tar -xzf bin.tar.gz

Übungsblatt 08

Aufgabe 5: find ~ -type d
find ~ -name "[!op]*"
find ~ -size +500
find ~ -size -100
find ~ -perm 0744
Aufgabe 7: chmod 000 SONSTIGES

Übungsblatt 09

Übungsblatt 10

```
#!/bin/bash

# Dateiname angegeben?
if [ -z "$1" ]
then echo "Sie muessen einen Dateinamen angeben."
exit 1
fi

# Datei vorhanden?
if [ ! -r "$1" ]
then echo "Datei $1 existiert nicht - wird angelegt."
touch $1
fi

# Endlosschleife
until false
do echo
echo " 1.: Adresse eintragen"
echo " 2.: Adresse suchen"
echo " 3.: Adressen anzeigen (alle)"
echo " 4.: Adresse loeschen"
echo " 5.: Programm beenden"
echo
echo -n "Bitte waehlen: "
read -n1 EINGABE
echo
case $EINGABE in
1) echo -n "Name: " ; read NAME
echo -n "Vorname: " ; read VORNAME
echo -n "Ort: " ; read ORT
echo -n "Telefon: " ; read TELEFON
echo "$NAME;$VORNAME;$ORT;$TELEFON" >> $1
;;

```

```
2) echo -n "Suchbegriff: " ; read SUCHBEGRIFF
GEFUNDENEZEILEN=$(grep "$SUCHBEGRIFF" $1)
for ZEILE in $GEFUNDENEZEILEN
do NAME=$( echo $ZEILE | cut -d\; -f1)
VORNAME=$( echo $ZEILE | cut -d\; -f2)
ORT=$( echo $ZEILE | cut -d\; -f3)
TELEFON=$( echo $ZEILE | cut -d\; -f4)
echo "$NAME, $VORNAME aus $ORT hat die \
Telefonnummer $TELEFON."
done
;;
3) for ZEILE in $(cat $1)
do NAME=$( echo $ZEILE | cut -d\; -f1)
VORNAME=$( echo $ZEILE | cut -d\; -f2)
ORT=$( echo $ZEILE | cut -d\; -f3)
TELEFON=$( echo $ZEILE | cut -d\; -f4)
echo "$NAME, $VORNAME aus $ORT hat die \
Telefonnummer $TELEFON."
done
;;
4) echo -n "Suchbegriff: " ; read SUCHBEGRIFF
ANDEREZEILEN=$(grep -v "$SUCHBEGRIFF" $1)
echo "$ANDEREZEILEN" > $1
;;
5) echo "Programm beendet"
exit 0
;;
*) echo "Falsche Eingabe."
;;
done
esac
```

Übungsblatt 12

Aufgabe 1.1:

```
#!/bin/bash
[ $# -ne 2 ] && echo "Fehler, zwei Parameter erforderlich." && exit 1
i=$1
until [ $i -gt $2 ]
do
    echo -n "$i "
    let i=$i+1
done
echo
```

Aufgabe 1.2:

```
#!/bin/bash
[ $# -ne 2 ] && echo "Fehler, zwei Parameter erforderlich." && exit 1
[ -d $1 ] || echo "Erster Parameter muss ein Verzeichniss sein." || exit 2
find $1 -type f -exec grep -l $2 {} 2>/dev/null \; 2>/dev/null | sort
```

Aufgabe 2:

```
#!/bin/bash
case $(basename $0) in
    aufgabe2)
        echo "gesperrt"
        exit 1
        ;;
    eins)
        echo "Erster Link"
        ;;
    zwei)
        echo "Zweiter Link"
        ;;
    drei)
        echo "Dritter Link"
        ;;
    *)
        echo "nicht definiert"
        ;;
esac
```

Aufgabe 3:

```
#!/bin/bash
prog=$(basename $0)
status=/tmp/$prog

case $1 in
    start)
        echo "Start."
        echo "Status: laeuft." > $status
        ( until false ; do echo -n "." ; sleep 1 ; done ) &
        ;;
    stop)
        echo "Stop."
        echo "Status: laeuft nicht." > $status
        killall $prog
        ;;
    status)
        if [ -f $status ]
        then cat $status
        else echo "lief noch nie"
        fi
        ;;
    *)
        echo "nicht definiert"
        ;;
esac
```

Übungsblatt 13

```
#!/bin/bash

# Dateiname angeben?
if [ -z "$1" ]
then dialog --msgbox "Sie muessen einen Dateinamen angeben." 7 50
    exit 1
fi

# Datei vorhanden?
if [ ! -r "$1" ]
then dialog --msgbox "Datei $1 existiert nicht - wird angelegt." 7 50
    touch $1
fi

# Endlosschleife
until false
do
WAHL=$( dialog --stdout --radiolist "Hauptmenü: bitte eine Aktion wählen:" 12 61 5 \
    "1" "Adresse eintragen" off \
    "2" "Adresse suchen" ON \
    "3" "Adressen anzeigen (alle)" off \
    "4" "Adresse loeschen" off \
    "5" "Programm beenden" off ) || break

case $WAHL in
1) NAME=$( dialog --stdout --inputbox "Name: " 9 30 )
VORNAME=$( dialog --stdout --inputbox "Vorname: " 9 30 )
ORT=$( dialog --stdout --inputbox "Ort: " 9 30 )
TELEFON=$( dialog --stdout --inputbox "Telefon: " 9 30 )
echo "$NAME;$VORNAME;$ORT;$TELEFON" >> $1
dialog --infobox --sleep 1 "eingetragen." 5 15
;;
2) SUCHBEGRIFF=$( dialog --stdout --inputbox "Suchbegriff: " 9 30 )
GEFUNDENEZEILEN="$(grep "$SUCHBEGRIFF" $1)"
AUSGABE=$( for ZEILE in $GEFUNDENEZEILEN
do NAME="$( echo $ZEILE | cut -d\; -f1)"
VORNAME="$( echo $ZEILE | cut -d\; -f2)"
ORT="$( echo $ZEILE | cut -d\; -f3)"
TELEFON="$( echo $ZEILE | cut -d\; -f4)"
echo "$NAME, $VORNAME aus $ORT hat die Telefonnummer $TELEFON."
done )
dialog --msgbox "$AUSGABE" 25 78
;;
3) AUSGABE=$( for ZEILE in $(cat $1)
do NAME="$( echo $ZEILE | cut -d\; -f1)"
VORNAME="$( echo $ZEILE | cut -d\; -f2)"
ORT="$( echo $ZEILE | cut -d\; -f3)"
TELEFON="$( echo $ZEILE | cut -d\; -f4)"
echo "$NAME, $VORNAME aus $ORT hat die Telefonnummer $TELEFON."
done )
dialog --msgbox "$AUSGABE" 25 78
;;
4) SUCHBEGRIFF=$( dialog --stdout --inputbox "Suchbegriff (löschen): " 9 30 )
ANDEREZEILEN="$( grep -v "$SUCHBEGRIFF" $1 )"
echo "$ANDEREZEILEN" > $1
dialog --infobox --sleep 1 "gelöscht." 5 15
;;
5) dialog --msgbox "ENDE." 7 50
    exit 0
;;
*) dialog --msgbox "Falsche Eingabe." 7 50
;;
esac
done
```