

Bash Skripting

- Benutzereingabe
- Parameter übergeben
- Zufall Random
- For-Schleife mit Zufallspasswort Random und While-Schleife
- Case - Kontrollstruktur
- For-Schleife
- Datei per Skript verschlüsseln

Benutzereingabe

```
#!/bin/bash

# Root oder nicht root?
if [[ $UID -ne 0 ]]
then
    echo "Das Skript läuft nicht mit Root-Rechten"
fi

# Den Login-Namen abfragen
read -p 'Bitte den Login-Namen eingeben: ' USERNAME

# Den vollständigen Namen abfragen
read -p 'Bitte den vollständigen Namen eingeben: ' COMMENT

# Das Passwort abfragen
read -p 'Bitte Das Paswort eingeben: ' PASSWORD

# Den Benutzer anlegen
useradd -m -c "$COMMENT" -s /bin/zsh $USERNAME

# Das Passwort setzen
echo "$SUERNAME:$PASSWORD | chpasswd # chpasswd muss verwendet werden, da passwd nicht über ein
skript funktioniert"
```

Usereingabe über Parameter

```
#!/bin/bash

# Root oder nicht root?
if [[ $UID -ne 0 ]]
then
```

```
echo "Das Skript läuft nicht mit Root-Rechten"
exit 1
fi

# Drei Parameter?
if [[ $# -ne 3 ]];then
    echo "Syntax: Adding-user-parm.sh USERNAME \"REAL NAME\" PASSWORD"
    exit 1
fi

USERNAME="$1"
COMMENT="$2"
PASSWORD="$3"

# Den Login-Namen abfragen
#read -p 'Bitte den Login-Namen eingeben: ' USERNAME

# Den vollständigen Namen abfragen
#read -p 'Bitte den vollständigen Namen eingeben: ' COMMENT

# Das Passwort abfragen
#read -p 'Bitte Das Paswort eingeben: ' PASSWORD

# Den Benutzer anlegen
useradd -m -c "$COMMENT" -s /bin/zsh $USERNAME

# Das Passwort setzen
echo "$SUERNAME:$PASSWORD | chpasswd # chpasswd muss verwendet werden, da passwd nicht über ein
skript funktioniert
```

Parameter übergeben

```
#!/bin/bash
```

```
# Parameter prüfen
```

```
if [[ $# -lt 1 ]];then
```

```
    echo "Syntax: parameter.sh USERNAME [USERNAME2]"
```

```
fi
```

```
# Parameter auslesen
```

```
echo "Name des Skripts: $(basename $0)" # mit $(basename) nur den Namen des Skripts anzeigen lassen  
$(dirname) würde den Pfad anzeigen
```

```
echo "Erster Parameter: $1"
```

```
echo "Zweiter Parameter: $2"
```

```
echo "Alle Parameter: $@"
```

```
echo "Du hast $# Parameter übergeben"
```

Zufall Random

```
$RANDOM
```

```
#Datum in Sekunden
```

```
date +%s
```

```
#Datum in Nanosekunden und Random wert
```

```
echo "$(date +%s%N)$RANDOM"
```

```
# Hashwert erstellen
```

```
echo "$(date +%s%N)$RANDOM" | sha512sum
```

```
# Reduzieren auf die ersten 8 Zeichen
```

```
echo "$(date +%s%N)$RANDOM" | sha512sum | head -c8
```

For-Schleife mit Zufallspasswort Random und While-Schleife

```
#!/bin/bash

# Wert in einer For-Schleife verarbeiten
for USER in $@
do
    useradd -m $USER
    PASSWORD=$(echo "$(date +%s%N$RANDOM)" | sha512sum | head -c8)
    echo "$USER:$PASSWORD" | chpasswd
    passwd -e $USER # Dadurch muss der User das Passwort bei der ersten Anmeldung ändern
    echo "User: $USER - Passwort: $PASSWORD"
done
```

Parameter aus einer Datei auslesen

```
#!/bin/bash

# Wert in einer For-Schleife verarbeiten
for USER in $(cat /home/eric/bin/user.txt)
do
    useradd -m $USER
    PASSWORD=$(echo "$(date +%s%N$RANDOM)" | sha512sum | head -c8)
    echo "$USER:$PASSWORD" | chpasswd
    passwd -e $USER # Dadurch muss der User das Passwort bei der ersten Anmeldung ändern
    echo "User: $USER - Passwort: $PASSWORD"
```

done

Mit Laufvariable

```
#!/bin/bash

for (( i=1; 1<11; i++ )); do
    echo $i
done

echo "Die schleife wurde beendet"
```

While Schleife

```
#!/bin/bash

# While-Schleife PoC

WEITER="j"

while [[ "$WEITER" 0 "j" ]]; do
    echo "While-Schleife wird ausgeführt"
    read -p "Weiter? j/n " WEITER
done

echo "Schleife beendet"
```

Case - Kontrollstruktur

Beispiel mit if

```
#!/bin/bash

# Programmverzweigung mit
if [[ "$#" -ne 2 ]]; then
    echo "Syntax: case.sh AKTION DIENST"
    exit 1
fi

if [[ "$1" = "start" ]];then
    echo "$2 wird gestartet"
    exit 0
fi

elif [[ "$1" = "stop" ]];then
    echo "$2 wird gestopt"
    exit 0
fi
```

Umsetzung mit Case

```
#!/bin/bash

if [[ "$#" -ne 2 ]]; then
    echo "Syntax: case.sh AKTION DIENST"
    exit 1
fi

# Verzweigung mit case
```

```
case $1 in
  start)
    echo "$2 wird gestartet"
    ;;
  stop)
    echo "$2 wird gestopt"
    ;;
  restart)
    echo "$2 wird neu gestartet"
    ;;
  status|state) # mit der pipe können wir mehrere möglichkeiten bieten diesen teil aufzurufen
    echo "Der Satus von $2 wird angezeigt"
    ;;
  *)
    echo "Der Parameter ist unbekannt"
    exit 1
    ;;
esac
```

For-Schleife

```
#!/bin/bash
```

```
echo "Die übergebenen Usernamen sind: $@"
```

```
# Alle Parameter (User) ausgeben
```

```
for USER in $@
```

```
do
```

```
    echo "User im Aktuellen For-Schleifen-Durchlauf: $USER"
```

```
done
```

Datei per Skript verschlüsseln

anschließend eine Datei mit `ccrypt` verschlüsselt:

```
#!/bin/bash

# Datei, die verschlüsselt werden soll
read -p "Bitte den Dateinamen eingeben, der verschlüsselt werden soll: " datei

# Passwort verdeckt eingeben
echo -n "Bitte das Passwort eingeben: "
read -s passwort
echo

# Datei mit ccrypt verschlüsseln
echo "$passwort" | ccrypt -e -k - "$datei"

# Überprüfen, ob die Verschlüsselung erfolgreich war
if [ $? -eq 0 ]; then
    echo "Die Datei wurde erfolgreich verschlüsselt."
else
    echo "Fehler bei der Verschlüsselung der Datei."
fi
```

Speichere dieses Skript in einer Datei, z.B. `encrypt.sh`, und mache es ausführbar mit:

```
chmod +x encrypt.sh
```

Dann kannst du es ausführen mit:

```
./encrypt.sh
```

Dieses Skript fordert den Benutzer auf, den Dateinamen und das Passwort einzugeben. Das Passwort wird verdeckt eingegeben (d.h. es wird nicht auf dem Bildschirm angezeigt). Anschließend wird die Datei mit `ccrypt` verschlüsselt.