

# Bash Scripting

## Ein Script erstellen

Eine Bash-Datei muss die Endung .sh haben

Aufbau der Bash:

```
#!/bin/bash  
  
echo Hello World!
```

Die Zeile `#!/bin/bash` sagt dem System, welche Shell sie verwenden werden, in diesem Fall die Bash-Shell.

## Script ausführbar machen

Damit jedes Skript ausgeführt werden kann muss es ausführbar gemacht werden.

```
chmod +x helloworld.sh
```

## Script ausführen

```
./helloworld.sh
```

# Programmbeispiele

Code	Kommentar
<code>\$(date +%A)</code>	ruft die Systemvariable ab, die den aktuellen Wochentag speichert

free

Der **free**-Befehl zeigt die Gesamtmenge des freien und verwendeten physischen Speichers und des Swap-Speichers im System an. **free -m** gibt z. B. die Informationen in Megabyte an.

sed

Der **sed**-Befehl öffnet einen Stream-Editor. Mit einem Stream-Editor werden Textumwandlungen in einem Eingabestream (Datei oder Eingabe aus einer Pipe) ausgeführt.

df

Der **df**-Befehl zeigt den Festplattenspeicherplatz des Dateisystems für alle Partitionen an. **df -h** ist wahrscheinlich der nützlichste Befehl (-h bedeutet, dass es für den Menschen lesbar ist).

adduser

Der **adduser**-Befehl fügt einen neuen Benutzer zum System hinzu. Auf ähnliche Weise fügt der Befehl **addgroup** eine neue Gruppe zum System hinzu.

top

Das **top**-Programm bietet eine dynamische Echtzeitansicht eines laufenden Systems. Es kann eine Zusammenfassung der Systeminformationen sowie eine Liste von Prozessen anzeigen.

deluser

Der Befehl **deluser** entfernt einen Benutzer aus dem System. Um dessen Dateien und Basisverzeichnis zu entfernen, müssen Sie die Option **-remove-home** hinzufügen.

uname-a

Der **uname**-Befehl gibt zusammen mit der Option **-a** alle Systeminformationen aus, einschließlich Computernamen, Kernelnamen, Version sowie einige weitere Infos.

delgroup

Der Befehl **delgroup** entfernt eine Gruppe aus dem System. Sie können keine Gruppe entfernen, die als primäre Benutzergruppe dient.

ps

Der **ps**-Befehl zeigt alle auf dem Computer ausgeführten Prozesse an. Die ps-Version unterscheidet sich geringfügig auf jedem Betriebssystem, aber sie führen alle dasselbe aus.

man man

Der **man man**-Befehl ruft den manuellen Eintrag für den **man**-Befehl auf, der für den Einstieg äußerst praktisch ist.

grep

Mit dem **grep**-Befehl können Sie in einer Reihe von Dateien nach einem bestimmten Suchmuster suchen und dann passende Zeilen ausgeben. Ein Beispiel wäre **grep Suchwort Dateiname**.

man intro

Der **man intro**-Befehl ist besonders nützlich. Er zeigt die Einführung in die Benutzerbefehle an, eine gut geschriebene, recht kurze Einführung in die Linux-Befehlszeile.



## A

<b>adduser</b>	Neuen Nutzer hinzufügen
<b>arch</b>	Prozessorarchitektur anzeigen
<b>awk</b>	Text in Dateien suchen und ersetzen

## B

<b>bc</b>	Rechenprogramm mit beliebiger Genauigkeit
-----------	---

## C

<b>cat</b>	Gibt Dateien hintereinander an die Standardausgabe weiter
<b>chdir</b>	Arbeitsverzeichnis wechseln
<b>chgrp</b>	Gruppenzugehörigkeit von Dateien ändern
<b>chroot</b>	Wurzelverzeichnis wechseln
<b>cksum</b>	CRC-Summe und Anzahl der Bytes ausgeben
<b>cmp</b>	Zwei Dateien vergleichen
<b>comm</b>	Zwei sortierte Dateien Zeile für Zeile vergleichen
<b>cp</b>	Datei(en) an einen anderen Speicherort kopieren
<b>crontab</b>	Befehl zu einem späteren Zeitpunkt ausführen lassen
<b>csplit</b>	Datei nach Kontext aufteilen
<b>cut</b>	Ausgewählte Felder jeder Zeile einer Datei ausschneiden

## D

<b>date</b>	Datum & Zeit anzeigen oder ändern
<b>dc</b>	Rechenprogramm

<b>dd</b>	Kopiert/konvertiert Dateien
<b>diff</b>	Unterschiede zwischen zwei Dateien anzeigen
<b>dirname</b>	Aus einem kompletten Pfadnamen das letzte Element entfernen

<b>du</b>	Geschätzte Speicherbelegung einer Datei
-----------	---

## E

<b>echo</b>	Nachricht auf dem Bildschirm anzeigen
<b>ed</b>	Zeilenorientierter Texteditor (edlin)
<b>egrep</b>	Zeilen in Dateien nach passenden erweiterten Ausdrücken durchsuchen
<b>env</b>	Umgebungsvariablen anzeigen, festlegen oder löschen
<b>expand</b>	Tabs in Leerzeichen konvertieren
<b>expr</b>	Ausdrücke auswerten

## F

<b>factor</b>	Primfaktoren ausgeben
<b>fdisk</b>	Konfiguration der Partitionstabelle in Linux
<b>fgrep</b>	Dateizeilen nach einem passenden feststehenden String durchsuchen
<b>find</b>	Nach Dateien suchen, die bestimmte Kriterien erfüllen
<b>fmt</b>	Absatz neu formatieren
<b>fold</b>	Text in eine bestimmte Breite umbrechen
<b>format</b>	Disketten oder Tapes formatieren

<b>fsck</b>	Dateisystem überprüfen und reparieren
-------------	---------------------------------------

## G

<b>gawk</b>	Text in Dateien finden und ersetzen
<b>grep</b>	Dateien nach Zeilen durchsuchen, die einem bestimmten Suchmuster entsprechen
<b>groups</b>	Gibt die Gruppen aus, denen ein Benutzer angehört
<b>gzip</b>	(De)Komprimiert angegebene Dateien

## H

<b>head</b>	Gibt den ersten Teil von Dateien aus
<b>hostname</b>	Systemname ausgeben oder ändern

## I

<b>id</b>	Benutzer- u. Gruppen-IDs ausgeben
<b>info</b>	Hilfe/Informationen
<b>install</b>	Dateien kopieren und Attribute setzen

## J

<b>join</b>	Zeilen eines gemeinsamen Feldes verknüpfen
-------------	--

## K

<b>kill</b>	Prozess beenden
-------------	-----------------

## L

<b>less</b>	Ausgabe seitenweise anzeigen
-------------	------------------------------

<b>ln</b>	Dateiverknüpfungen herstellen
<b>locate</b>	Dateien finden
<b>logname</b>	Aktuellen Anmeldenamen anzeigen
<b>lpc</b>	Zeilendruckersteuerung
<b>lpr</b>	Dateien drucken
<b>lprm</b>	Aufträge in der Druckwarteschlange löschen

## M

<b>man</b>	Handbuch anzeigen
<b>mkdir</b>	Neues Verzeichnis erstellen
<b>mkfifo</b>	FIFOs (benannte Pipes) erstellen
<b>mknod</b>	Block- oder Zeichen-dateien erstellen
<b>more</b>	Ausgabe seitenweise anzeigen
<b>mount</b>	Dateisystem einrichten

## N

<b>nice</b>	Priorität eines Befehls/ einer Aufgabe festlegen
<b>nl</b>	Zeilen nummerieren
<b>nohup</b>	Befehl trotz Systemabmeldung ausführen

## P

<b>passwd</b>	Benutzerpasswort ändern
<b>paste</b>	Dateizeilen zusammenführen
<b>pathchk</b>	Übertragbarkeit des Dateinamens prüfen
<b>pr</b>	Textdateien zum Drucken konvertieren
<b>printcap</b>	Datenbank für die Druckerfähigkeit
<b>printenv</b>	Umgebungsvariablen ausgeben
<b>printf</b>	Daten formatieren und ausgeben

## Q

<b>quota</b>	Festplattennutzung und -begrenzung anzeigen
<b>quotacheck</b>	Dateisystem nach Festplattennutzung überprüfen

<b>quotactl</b>	Festplattenquoten festlegen
-----------------	-----------------------------

## R

<b>ram</b>	RAM-Speichergerät
<b>rcp</b>	Dateien zwischen zwei Geräten kopieren
<b>rm</b>	Datei(en) löschen
<b>rmdir</b>	Verzeichnis(se) löschen
<b>rpm</b>	Remote Package Manager
<b>rsync</b>	Fernkopieren von Dateien (Dateibäume synchronisieren)

## S

<b>screen</b>	Terminal-Fenstermanager
<b>sdiff</b>	Zwei Dateien interaktiv zusammenführen
<b>select</b>	Tastatureingabe übernehmen
<b>seq</b>	Zahlenfolgen ausgeben
<b>shutdown</b>	Linux herunterfahren/ neu starten
<b>sleep</b>	Für eine festgelegte Zeit unterbrechen
<b>sort</b>	Textdateien sortieren
<b>split</b>	Datei in festgelegte Größen aufteilen
<b>su</b>	Benutzer-ID wechseln
<b>sum</b>	Prüfsumme einer Datei

<b>sum</b>	Prüfsumme einer Datei ausgeben
<b>symlink</b>	Neuen Dateinamen erstellen
<b>sync</b>	Festplatte mit Arbeitsspeicher synchronisieren

## T

<b>tac</b>	Dateien in umgekehrter Reihenfolge verketteten & schreiben
<b>tail</b>	Letzten Teil einer Datei ausgeben
<b>tar</b>	Archivierungsprogramm
<b>tee</b>	Ausgabe auf mehrere Dateien weiterleiten
<b>test</b>	Bedingungsausdruck auswerten
<b>time</b>	Nutzung der Programmressourcen bemessen

<b>touch</b>	Datei-Zeitstempel ändern
<b>top</b>	Laufende Prozesse auflisten
<b>traceroute</b>	Route zum Host verfolgen
<b>tr</b>	Zeichen übersetzen, schmälern oder löschen
<b>tsort</b>	Topologische Sortierung

## U

<b>umount</b>	Gerät auswerfen
<b>unexpand</b>	Leerzeichen in Tabs konvertieren
<b>uniq</b>	Dateien ohne doppelte Zeilen ausgeben
<b>units</b>	Einheiten von einer Skala zur nächsten konvertieren
<b>unshar</b>	Shell-Archivskripte entpacken
<b>useradd</b>	Benutzerkonto erstellen
<b>usermod</b>	Benutzerkonto ändern
<b>users</b>	Angemeldete Nutzer auflisten

## V

<b>vdir</b>	Ausführliche Liste der Verzeichnisinhalte ('ls -l -b')
-------------	--

## W

<b>watch</b>	Regelmäßiges ausführen/ anzeigen eines Programms
--------------	--

<b>wc</b>	Anzahl der Bytes, Wörter und Zeilen ausgeben
<b>whereis</b>	Alle bekannten Instanzen eines Befehls anzeigen
<b>which</b>	Nach Programmdatei im Benutzerpfad suchen
<b>who</b>	Alle aktuell angemeldeten Benutzernamen ausgeben
<b>whoami</b>	ID und Name des aktuellen Benutzers ausgeben

## X

<b>xargs</b>	Weitergabe von Argumenten von einem Befehl zum anderen
--------------	--

## Y

<b>yes</b>	Gibt einen String bis zum Abbruch aus
------------	---------------------------------------

# Variable übergeben

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

```
echo Hello $1
```

`./hello.sh Hermann` gibt Hallo Hermann auf dem Bildschirm aus

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

```
firstname=$1
```

```
surname=$2
```

```
echo Hello $firstname $surname
```

Bei Variablen wird zwischen Name und '=' keine Leerzeichen gelassen!

# Klammern

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

```
firstnumber=$1
```

```
secondnumber=$2
```

```
echo The sum is $((firstnumber+secondnumber))
```

# Input

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

```
echo -n "Hello, what is your name? " #-n bewirkt, dass die Eingabe nicht in einer neuen Zeile stattfindet
```

```
read firstname
```

```
echo -n "Thank you, and what is your surname? "
```

```
read surname
```

```
clear
```

```
echo Hello $firstname $surname, how are you today?
```

# If, then und Else

```
#!/bin/bash

echo -n "Hello, what is your Name "
read firstname
echo -n "And what is your surname? "
read surname
clear
if [ "$firstname" == "Hermann" ] && [ "$surname" == "Pelzer" ]
then echo "Awesome name," $firstname $surname
else echo Hello $firstname $surname, how are you today?
fi
```

# While-Schleife

```
#!/bin/bash

count=0

while [ $count -lt 100 ];do
echo $count
let count=count+1
done
```

# For-Schleife

```
#!/bin/bash

for count in {0..100}; do
echo $count
let count=count+1
done
```

# Choice

```

#!/bin/bash

auswahl = 4

echo "1. Tails"
echo "2. Is"
echo "3. Awesome"
echo -n "Please choose an option (1, 2, or 3) "

while [ $choice -eq 4 ] ; do

read choice

if [ $choice -eq 1 ] ; then
    echo "You have chosen: Tails"
else
    if [ $choice -eq 2 ] ; then
        echo "You have chosen: Is"
    else
        if [ $choice -eq 3 ] ; then
            echo "You have chosen: Awesome"
        else
            echo "Pleas make a choise between 1 to 3"
            echo "1. Tails"
        fi
    fi
    echo "2. Is"
    echo "3. Awesome"
    echo -n "Please choose an option (1, 2, or 3) "
    choice=4
fi
fi
done

```

---

Revision #6

Created 17 April 2023 08:55:39 by Hermann

Updated 17 April 2023 11:07:11 by Hermann