

EIB v1.1

EIB, MIT, <https://github.com/eisenwinter/fh-hgb-stuff>
Jan Caspar, Aktualisiert 9. Februar 2017

Bash Commands

date	Datum und Uhrzeit
uptime	Uptime
whoami	User
man command	Manual für command
!!	letzten Befehl wiederholen

IO Redirection

cmd < file	Eingabe für cmd von file
cmd1 < (cmd2)	Ausgabe von cmd2 als Eingabe für cmd1
cmd > file	Ausgabe von cmd in file
cmd > /dev/null	Ausgabe (stdout) verwerfen
cmd >> file	Ausgabe von cmd in file anhängen
cmd 2> file	stderr (Errorouput) von cmd in file
cmd 1>&2	stdout und stderr in gleiche Ausgabe
cmd 2>&1	stderr und stdout in gleiche Ausgabe
cmd &> file	ganze Ausgabe von cmd in file

cmd ist ein beliebiger Command

Pipes

cmd1 cmd2	stdout von cmd1 in cmd2
cmd1 & cmd2	stderr von cmd1 in cmd2

Zusammenhängen

cmd1 ; cmd2	Zuerst cmd1 dann cmd2
cmd1 && cmd2	cmd2 wenn cmd1 erfolgreich
cmd1 cmd2	cmd2 wenn cmd1 nicht erfolgreich
cmd &	cmd in einer Unterschell

Verzeichnisse

pwd	derzeitiges Verzeichnis
mkdir dir	Verzeichnis dir anlegen
cd dir	in Verzeichnis dir wechseln
cd ..	ein Verzeichnis höher
ls	Verzeichnisinhalt anzeigen

LS Argumente im Detail

-a	Show all (including hidden)
-R	Recursive list
-r	Reverse order
-t	Sort by last modified
-S	Sort by file size
-l	Long listing format
-1	One file per line
-m	Comma-separated output
-Q	Quoted output

Dateioperationen

touch file1	file1 anlegen
cat file1 file2	Ausgabe zusammenhängen
less file1	Anzeigen mit Scrollfunktion
file file1	Typ von file1 anzeigen
cp file1 file2	Kopiert file1 nach file2
mv file1 file2	Verschiebt file1 nach file2 (auch umbenennen)
rm file1	löscht file1
rm -r /dir	löscht Verzeichnis dir und alle Dateien / Unterordner darin
head file1	Die ersten 10 Zeilen aus file1
tail file1	Die letzten 10 Zeilen aus file1

Berechtigungen

Setzen

chmod 775 file	Berechtigung von file auf 775 ändern
chmod -R 600 folder	Rekursiv Berechtigung für Order folder auf 600 setzen
chown user:group file	Besitzer und Gruppe ändern
chmod g-r	Gruppe Leserechte einziehen
chmod u+x	User Ausführrechte geben
umask 007	Entzieht Berechtigung für alle nächsten anlegte Dateien und Verzeichnisse.

Beispiel umask

777 → 730

umask 003
touch file
mkdir

Nummern

Erste Zahl ist Besitzer Berechtigung, Zweite ist Gruppenberechtigung und dritter ist Jeder.

Berechnet wird die Zahl in dem man die unten stehenden Zahlen addiert. (subtrahieren bei umask)

4	read (r)
2	write (w)
1	execute (x)

Textfiles verarbeiten

tr

tr -s " "
Bereinigt überflüssige Leerzeichen

cut

cut schneidet Spalten oder Felder aus Textfiles aus. Die Spalten oder Felder sind ab 1 nummeriert, es können spezifische Trennzeichen verwendet werden. Standard Trennzeichen ist <TAB>
Beispiel
cut -f4 -d " "
Schneidet die 4te Spalte aus, geht von Leerzeichen getrennten Spalten aus.

sort

Sortiert die Ausgabe von Text.

sort -k9 ls.txt	Sortiert list nach der 9ten Spalte. Nicht numerisch.
sort -t ";" -nk2,5 -k9	Sortiert nach Spalte 2 gretrennt mit , numerisch, danach nach Spalte 9 nicht numerisch.
sort -r	Sortiert nicht numerisch, absteigend.
sort -b	Leerzeichen am Anfang ignorieren
sort -f	Behandelt kleine Buchstaben wie große Buchstaben

uniq

Mehrfach vorkommende gleiche Zeilen auf eine zu reduzieren.
Funktioniert **nur** auf bereits sortierte Files, sonst werden Textzeilen mit gleichem Inhalt nicht gefunden.

tail

Die letzten 10 Zeilen einer Datei / Ausgabe.
Mit Argument -nl werden statt 10 die letzten n Zeilen ausgegeben

head

Wie tail nur die ersten 10 Zeilen.

wc

wordcount, dient zum Zählen von Wörtern, Zeichen und Bytes in Textdateien.
Folgende Optionen sind möglich

-l	zählt die Zeilen in der Datei
-c	zählt die Bytes in der Datei
-L	gibt die Länge der längsten Zeile aus
-w	zählt die Worte in der Datei

Benutzerverwaltung

Informationen über den aktuellen Benutzer:

id
whoami

Benutzer hinzufügen:

sudo adduser <benutzername>
Mit diesem Kommando wird automatisch eine neue Gruppe mit dem Namen
<benutzername> als primäre Gruppe für den neuen Benutzer angelegt und der neue
Benutzer als einziges Mitglied eingetragen. Weiters wird sein Home Verzeichnis angelegt und der
Inhalt von /etc/skel in dieses Home-Verzeichnis kopiert.

Benutzer löschen ohne Homeverzeichnis mitzulöschen

sudo deluser <benutzername>

Benutzer inkl. Homeverzeichnis löschen

sudo deluser --remove-home <benutzername>

Neuen Benutzer hinzufügen und seine primäre Gruppe bestimmen:

sudo adduser <benutzername> --ingroup <gruppenname>

Bestehenden Benutzer einer weiteren Gruppe hinzufügen:

sudo usermod -aG <gruppenname> <benutzername>

Primäre Gruppe eines Benutzers ändern:

sudo usermod -g <gruppenname> <benutzername>

Benutzer aus einer Gruppe entfernen:

sudo deluser <benutzername> <gruppenname>

Eine neue Gruppe erstellen:

sudo addgroup <gruppenname>

Eine bestehende Gruppe löschen:

sudo delgroup <gruppenname>

Benutzernamen ändern:

usermod -l <neuerName> <alterName>
Allerdings nur möglich, wenn der zu ändernde Benutzername gerade nicht am System angemeldet ist.

Passwort ändern:

Eigenes Passwort ändern:
passwd
Passwort eines anderen Benutzers ändern:
sudo passwd <benutzername>

root Account aktivieren/deaktivieren:

Durch Vergabe eines Passwortes:
sudo passwd root
root Account sperren (lock):
sudo passwd -l root
root Account entsperren (unlock):
sudo passwd -u root

Einen Benutzer zum „sudoer“ machen, ohne Änderung der Defaultdatei /etc/sudoers.

- Schritt 1: Anlegen des neuen Benutzers
sudo adduser <benutzername>
- Schritt 2: Einfügen des Benutzers in die Gruppe sudo:
sudo usermod -aG sudo <benutzername>
- Schritt 3: Testen, ob's funktioniert (zB. mit):
su <benutzername> # switch user, öffnet neue Shell
cat /etc/shadow (sollte File anzeigen)
exit # zur alten Shell zurückkehren
- Schritt 4: Löschen eines Benutzers aus der Gruppe sudo:
sudo deluser <benutzername> sudo

Bashscripting

Shebangline

Allgemein

```
#!/bin/sh
```

Implizit bash

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

Executable chmod

```
chmod u+x shellsript.sh
```

Pattern Matching

*	Entspricht beliebiger Eingabe
?	Entspricht einem Buchstaben
[AaBbCc]	Entspricht einem der Buchstaben
[^RGB]	Entspricht NICHT einem der Buchstaben
[a-g]	Entspricht einem Buchstaben zwischen a und g

Escaping

\c	Gibt c aus.
'cmd'	Gibt den Rückgabewert von Befehl cmd aus. (Backquotes)
"whatever"	Whatever sofern es nicht \$whatever und 'whatever' gibt.
'whatever'	Gibt ausschließlich whatever aus

Standard Shellvariablen

\$0	Name des Shell Scripts
\$1	Erster übergebener Parameter (analog \$2,\$3...)
\$#	Die Anzahl der übergebenen Parameter
\$*	Alle Parameter die Übergeben wurden
\$-	Optionen mit der die Shell aufgefunden wurde
\$?	Exit Code des letzten Commands
\$\$	Prozess ID der Shell
\$RANDOM	Zufallsnummer

Berechnungen

```
z = 0
z=$(( $z+3 ))
```

If / test

```
if [ "$x" -lt "$y" ];
then
    #irgendwas
else
    #iwas sonst
fi
```

Bediengungen

Numerisch

lt	weniger als
gt	größer als
eq	gleich
ne	nicht gleich
ge	größer gleich
le	kleiner gleich

Datei

nt	neuer als
d	ist Verzeichnis
f	ist Datei
x	ist ausführbar
r	ist lesbar
w	ist schreib-bar

String

=	gleich
z	hat Länge o
n	hat Länge > o

Logik

&&	logisches Und
	logisches Oder
!	logisches Nicht

Beispiele

```
if [ $# -eq 0 ]
then
    echo "No_arguments_defined"
```

```
else
    echo "Number_of_arguments_is:_$#"
fi

if test $# -eq 0
then
    echo "No_arguments_defined"
else
    echo "Number_of_arguments_is:_$#"
fi
```

```
#prüfen ob folder existiert
#sonst anlegen
DIR = ./folder
if [ ! -d $DIR ]
then
    mkdir $DIR
```

```
else
    echo "Already_exists"
fi
```

Schleifen

```
for arg in [list]
do
    echo $arg
done
```

```
#wörter einzeln aus file in $1
for x in `cat $1`
do
    echo $x
done
```

```
X=""
while [ "$X" != "end" ]
do
    echo -n "_Eingabe:_"
    read X
    echo "Ausgabe:_$X"
done
```

```
#textfile zeilenweise auslesen mit while
while read line
do
    echo $line
done < txt.txt
```

```
# Schleife bricht ab, wenn N > 10
N=1
while [ $N -le 10 ]
do
    echo "N_hat_jetzt_den_Wert_$N"
    N=`expr $N + 1`
done
```

Eingabe

```
read <Variable>

#!/bin/sh
echo "Whats_your_name?_"
read MY_NAME
echo "Hello_$MY_NAME"
```