

# Shell-Praxis (Teil 1)




**BASH**  
THE BOURNE-AGAIN SHELL



# Inhaltsverzeichnis

- [Vorbemerkungen](#)
- [Übersicht](#)
- `bash` [-Prompt](#)
- [Das aktuelle Verzeichnis ermitteln:](#) `pwd`
- [Variablen](#) `$USER`, `_` `$HOME` [und](#) `~`
- `sudo` [\\_\(superuser do\)](#)

- Pakete aktualisieren mit `apt`
- Pakete installieren mit `apt`
- Datum und Kalender: `date` und `cal`
- Befehlswiederholung
- `Ctrl+C`, `Shift+Ctrl+C` und `Shift+Ctrl+V`
- ASCII-Kunst mit `cowsay`
- Sinnsprüche und Zitate mit `fortune`

- [Lesen von der Standardeingabe](#)
- [Kommandos verbinden mit Pipes](#) 
- [Kommandosubstitution](#)
- [Aliase](#)
- [Man-Pages](#)
- [Shutdown und Reboot](#)

# Vorbemerkungen

Die folgende Beispiel-Sitzung will anhand von einfachen Beispielen die Arbeit mit der Shell (Bash) im Terminal vorstellen. Der Schwerpunkt liegt auf der praktischen Anwendung der Shell und der wichtigsten Kommandos. Erklärungen und Erläuterungen sind auf das notwendige Minimum beschränkt.

# Übersicht

In diesem Foliensatz werden die Grundlagen der Arbeit mit dem Terminal (oder einem Terminal-Emulator) vorgestellt.

Wenn man sich am Terminal anmeldet oder einen Terminal-Emulator startet, wird im Terminal eine Shell gestartet. Die Shell ist ein Programm, das Befehle entgegen nimmt und ausführt. Bei Linux-Systemen ist die Standard-Shell meistens die Bash (Bourne Again Shell). Die Bash gibt zu Beginn und nach jeder Kommandoausführung einen Prompt aus, der den Benutzer auffordert, ein Kommando einzugeben.

# Der **bash**-Prompt

```
hermann@debian:~$
```

Der Prompt besteht aus dem Benutzernamen ( **hermann** ), einem **@** - Zeichen, dem Hostnamen des Computers ( **debian** ), einem Doppelpunkt und dem aktuellen Verzeichnis ( **~** ).

Hinter dem Prompt blinkt ein Cursor, der anzeigt, dass die Shell auf Eingaben wartet. Der Benutzer kann ein Kommando eingeben und mit der Return-Taste bestätigen.

# Das aktuelle Verzeichnis ermitteln: `pwd`

- Nach dem Öffnen einer Terminal-Sitzung "steht" die Shell im Heimat-Verzeichnis des Benutzers.
- Das aktuelle Verzeichnis wird im Prompt angezeigt.
- Es kann auch mit dem Kommando `pwd` (print working directory) ermittelt und ausgegeben werden.

```
hermann@debian:~$ pwd  
/home/hermann
```



# Variablen `$USER`, `$HOME` und `~`

Die beiden Umgebungsvariablen `$USER` und `$HOME` werden beim Anmelden automatisch gesetzt.

- `$USER` enthält den Benutzernamen.
- `$HOME` enthält den Pfad zum Heimat-Verzeichnis des Benutzers.
- `~` ist eine Abkürzung für `$HOME`.

Mit dem Befehl `echo` werden die einfach die Argumente dieses Kommandos ausgegeben, d.h. in die Standardausgabe geschrieben.

## `$USER`, `$HOME` und `~` (Beispiele)

```
hermann@debian:~$ echo $USER
hermann
hermann@debian:~$ echo Ich bin $USER.
Ich bin hermann.
hermann@debian:~$ echo $HOME
/home/hermann
hermann@debian:~$ echo Hier wohne ich: $HOME
Hier wohne ich: /home/hermann
hermann@debian:~$ echo Hier wohne ich: ~
Hier wohne ich: /home/hermann
hermann@debian:~$ echo ~
/home/hermann
```

# **sudo** (superuser do)

- Befehle, die Eingriffe in das System sind (z.B. Installation von Software, Änderung von System-Konfigurationsdateien), benötigen zur Ausführung Superuser-Rechte.
- Der Befehl **sudo** (superuser do) wird vor einem Kommando eingegeben, um dieses Kommando mit Superuser-Rechten auszuführen.
- Die **sudo**-Berechtigung für den bei der Installation angelegten Benutzer steht nur dann auf Anhieb zur Verfügung, wenn bei der Installation das root-Passwort leer gelassen wurde.

- `sudo` fragt nach dem Passwort des Benutzers und führt erst nach erfolgreicher Passworteingabe das betreffende Kommando aus.
- Eine sudo-Session bleibt für eine gewisse Zeit (Standard: 5 Minuten) aktiv, so dass das Passwort erst nach Ablauf dieser Zeit erneut eingegeben werden muss.
- Der `sudo`-Session-Timeout kann sich bei verschiedenen Linux-Distributionen unterscheiden und ist konfigurierbar.

## `sudo` -Berechtigung nachträglich einrichten

Voraussetzung: Bei der Installation von Debian wurde das root-Passwort (anders als empfohlen) vergeben.

### Schritte:

1. Wechsel in das root-Konto: `su -`
2. root-Passwort eingeben, das bei der Installation vergeben wurde
3. Hinzufügen des Benutzers zur Gruppe `sudo`:

```
usermod -aG sudo <username>
```

# Pakete aktualisieren mit `apt`

- `apt update` aktualisiert die Paketlisten.
- Mit `apt list --upgradable` kann man sich die Pakete anzeigen lassen, die aktualisiert werden können.
- `apt upgrade` aktualisiert die Pakete, für die eine neuere Version verfügbar ist.

- `apt update` und `apt upgrade` müssen mit `sudo` ausgeführt werden, da sie einen administrativen Eingriff in das System darstellen.
- Die Befehle werden gerne mit `&&` kombiniert, um sie nacheinander auszuführen. So wird der zweite Befehl aber nur ausgeführt, wenn der erste erfolgreich war.

```
hermann@debian:~$ sudo apt update && sudo apt upgrade
Hit:1 http://deb.debian.org/debian bookworm InRelease
Hit:2 http://security.debian.org/debian-security bookworm-security InRelease
Hit:3 http://deb.debian.org/debian bookworm-updates InRelease
Hit:4 http://ftp.debian.org/debian bookworm-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```



# Pakete installieren mit `apt`

- `apt install <packagename>` installiert ein Paket.
- `apt remove <packagename>` deinstalliert ein Paket.
- Auch diese Befehle müssen mit `sudo` ausgeführt werden, da sie einen administrativen Eingriff in das System darstellen.

## Befehl `cal` aus Paket `ncal` nachinstallieren

```
hermann@debian:~$ sudo apt install ncal
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  ncal
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 19.7 kB of archives.
After this operation, 59.4 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bookworm/main amd64 ncal amd64 12.1.8 [19.7 kB]
Fetched 19.7 kB in 1s (28.9 kB/s)
Selecting previously unselected package ncal.
(Reading database ... 9538 files and directories currently installed.)
Preparing to unpack .../archives/ncal_12.1.8_amd64.deb ...
Unpacking ncal (12.1.8) ...
Setting up ncal (12.1.8) ...
```

# Datum und Kalender: `date` und `cal`

- `date` zeigt das aktuelle Datum und die aktuelle Uhrzeit.
- `cal` zeigt den Kalender des aktuellen Monats.
- `cal <year>` zeigt den Kalender des angegebenen Jahres.
- `cal` muss auf Debian-Systemen ggf. nachinstalliert werden (Paket: `ncal`)

```
hermann@debian:~$ date
Sun Nov  3 08:02:27 PM CET 2024
hermann@debian:~$ cal
    November 2024
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
```

## Paket **nca1** wieder deinstallieren

```
hermann@debian:~$ sudo apt remove ncal
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages will be REMOVED:
  ncal
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.
After this operation, 59.4 kB disk space will be freed.
Do you want to continue? [Y/n]
(Reading database ... 9546 files and directories currently installed.)
Removing ncal (12.1.8) ...
```

Installieren Sie das Paket **nca1** wieder, sodass weiter verfügbar ist.

# Befehlswiederholung

- Mit den Pfeiltasten `↑` und `↓` kann man durch die Befehlshistorie scrollen.
- Mit dem Kommando `history` werden die letzten Befehle mit einer Nummerierung angezeigt.
- Mit `!!` wird der letzte Befehl wiederholt.
- Mit `!n` wird der Befehl mit der Nummer `n` wiederholt.
- Mit `!string` wird der letzte Befehl, der mit `string` beginnt, wiederholt.

```
hermann@debian:~$ history
```

```
1  pwd
```

```
2  ls -l
```

```
...
```

```
71  date
```

```
72  cal
```

```
73  sudo apt install ncal
```

```
74  cal
```

```
75  sudo apt remove ncal
```

```
76  history
```

```
hermann@debian:~$ !71
```

```
date
```

```
Sun Nov  3 08:25:55 PM CET 2024
```

```
hermann@debian:~$ !d
```

```
date
```

```
Sun Nov  3 08:26:06 PM CET 2024
```

## **Ctrl+C**, **Shift+Ctrl+C** und **Shift+Ctrl+V**

- Mit **Ctrl+C** wird ein laufendes Kommando abgebrochen.

Beispiel: Sie führen einen **ping**-Befehl aus, der sich niemals beendet.  
Mit **Ctrl+C** können Sie ihn abbrechen.



- Mit `Shift+Ctrl+C` wird der im Terminal-Emulator markierte Text in die Zwischenablage kopiert.
- Mit `Shift+Ctrl+V` wird der Text aus der Zwischenablage an der aktuellen Cursor-Position in den Terminal-Emulator eingefügt.
- Copy & Paste funktioniert nur in einem Emulator, nicht in einem echten Terminal.

# ASCII-Kunst mit **cowsay**

```
hermann@debian:~$ sudo apt install cowsay
Reading package lists... Done
...
Setting up cowsay (3.04+dfsg2-7) ...
hermann@debian:~$ cowsay "Hallo, ich bin ein Linux-Pinguin!"

-----
< Hallo, ich bin ein Linux-Pinguin! >
-----

      ^__^
      (oo)\_______
          (__)\       )\/\
              ||----w |
              ||     ||
```

```
hermann@debian:~$ cowsay -f elephant "Hallo, ich bin ein Linux-Pinguin!"
```

```
-----  
< Hallo, ich bin ein Linux-Pinguin! >  
-----
```

```
\      /\  _ _  /\  
 \    //  \ /   \ /  \\  
  ((    0 0    ))  
  \ \  /      \  //  
   \ /   | |   \ /  
    |    | |   |  
    |    | |   |  
    |    o    |  
    |    |    |  
   |m|    |m|
```

```
hermann@debian:~$ cowsay -f tux "Hallo, ich bin ein Linux-Pinguin!"
```

```
-----  
< Hallo, ich bin ein Linux-Pinguin! >  
-----
```

```
  \
   \
      .--.
    |o_o |
    |:_/ |
  //     \ \
  (|       |)
 /'\_ _   _/\`\'
 \_____)=(_____/
```

- Mit der Option `-f <cowfile>` geben sie an, welches ASCII-Art-File verwendet werden soll.
- Die Cow-Files liegen im Verzeichnis `/usr/share/cowsay/cows`.
- Mit der Option `-l` werden alle verfügbaren Cow-Files angezeigt.

```
hermann@debian:~$ cowsay -l
Cow files in /usr/share/cowsay/cows:
apt bud-frogs bunny calvin cheese cock cower daemon default dragon
dragon-and-cow duck elephant elephant-in-snake eyes flaming-sheep fox
ghostbusters gnu hellokitty kangaroo kiss koala kosh luke-koala
mech-and-cow milk moofasa moose pony pony-smaller ren sheep skeleton
snowman stegosaurus stimpy suse three-eyes turkey turtle tux unipony
unipony-smaller vader vader-koala www
```

# Sinnsprüche und Zitate mit `fortune`

- Das Programm `fortune` gibt nach dem Zufallsprinzip Sinnsprüche oder Zitate aus, bei jedem Aufruf einen anderen.
- Auch hier muss das Paket `fortune` installiert werden, damit das Programm `fortune` benutzt werden kann.

```
hermann@debian:~$ sudo apt install fortune
...
hermann@debian:~$ fortune
Your business will go through a period of considerable expansion.
hermann@debian:~$ fortune
You will step on the night soil of many countries.
hermann@debian:~$ fortune
Do not go around saying the world owes you a living.
The world owes you nothing. It was here first.
-- Mark Twain
```

# Lesen von der Standardeingabe

`cowsay` liest - wie viele andere Kommandos auch - von der Standardeingabe.

Wenn man `cowsay` keine Nachricht als Argument gibt, dann liest `cowsay` von der Standardeingabe.

Gibt man hinter dem Prompt nur `cowsay` ein, dann wartet das Programm auf die Benutzereingabe. Nun kann man beliebigen auch mehrzeiligen Text eintippen, den man mit `Ctrl+D` (am Zeilenanfang) abschließen muss. Nach dem Abschluss der Eingabe liest das Programm den eingegebenen Text und verarbeitet ihn.




```
hermann@debian:~$ cowsay
Diesen Text frisst cowsay von STDIN.
Es duerfen auch mehrere Zeilen sein.
Nach der 3. Zeile druecke ich auf Ctrl+D.

-----
/ Diesen Text frisst cowsay von STDIN. Es \
| duerfen auch mehrere Zeilen sein. Nach |
\ der 3. Zeile druecke ich auf Ctrl+D.    /
-----

      ^__^
      (oo)\_______
          (--)\/      )\/\
              ||----w |
              ||     ||
```

# Kommandos verbinden mit Pipes

- Werden zwei Kommandos mit einem -Symbol verbunden, so wird die Standardausgabe des ersten Kommandos an die Standardeingabe des zweiten Kommandos weitergeleitet.
- So können auch drei, vier oder mehr Kommandos miteinander zu einer sog. Pipeline verbunden werden.

```
hermann@debian:~$ date | cowsay
```

```
-----  
< So 3. Nov 23:28:47 CET 2024 >  
-----
```

```
      ^__^  
      (oo)\_____  
      (--)\\       )\\/\  
           ||----w |  
           ||     ||
```

```
hermann@debian:~$ fortune | cowsay -f bunny
```

```
-----  
/ Always do right. This will gratify some \  
| people and astonish the rest.           |  
|                                           |  
\ -- Mark Twain                          /  
-----
```

```
\  
 \  
  \  
   \  
    \  
   / \  
  (  )  
 .(  o  ).
```

# Kommandosubstitution

- Mit `$(command)` wird das Kommando `command` ausgeführt und die Ausgabe des Kommandos an der Stelle der Kommandosubstitution `$(command)` eingesetzt.
- Bei `cowsay $(date)` wird das Kommando `date` ausgeführt und die Ausgabe von `date` an die Stelle von `$(date)` eingesetzt. So kann in diesem Fall dasselbe erreicht werden wie mit der Pipe-Konstruktion `date | cowsay`.

```
hermann@debian:~$ cowsay $(date)
```

```
-----  
< So 3. Nov 23:39:16 CET 2024 >  
-----
```

```
      ^__^  
      (oo)\_____  
      (--)\\       )\\/\  
           ||----w |  
           ||     ||
```

# Aliase

- Mit dem Befehl `alias` können Aliase definiert werden.
- Ein Alias ist ein Kurzname für ein Kommando oder eine Kommandofolge.
- Ein Alias wird so definiert: `alias aliasname='kommando'`
- Empfehlung: Schließen Sie die Alias-Definition rechts vom `=`-Zeichen in einfache Anführungszeichen ein. (Begründung an anderer Stelle)
- Ein Alias wird in der aktuellen Shell-Sitzung definiert und ist nur in dieser Sitzung gültig.

```
hermann@debian:~$ alias cowdate='cowsay $(date)'
```

```
hermann@debian:~$ cowdate
```

```
-----  
< So 3. Nov 23:36:48 CET 2024 >  
-----  
      ^__^  
      (oo)\_____  
      (__)\\       )\\/\  
           ||----w |  
           ||     ||
```



```
hermann@debian:~$ alias bunnyfortune='fortune | cowsay -f bunny'
```

```
hermann@debian:~$ bunnyfortune
```

```
-----  
/ You will be singled out for promotion \  
\ in your work.                          /  
-----
```

```
\  
 \  
  \  
   \  
  /\   
  ( )   
  ( o ).
```

```
hermann@debian:~$ cocksay
cocksay: Befehl nicht gefunden.
hermann@debian:~$ alias cocksay='cowsay -f cock'
hermann@debian:~$ cocksay Quaaaaak
```

```
-----
< Quaaaaak >
-----
      \
       \  /\ /\
        \  /
         | 0 >>
         |---|
    __(((_<|   |
   (         |
  (-----)   |
   |         |
   |         |
   /\        /\
```

## Aliase dauerhaft verfügbar machen in `$HOME/.bash_aliases`

- Wird ein Alias hinter dem `bash`-Prompt definiert, dann gilt er nur in der aktuellen Shell-Sitzung.
- Soll der Alias dauerhaft verfügbar gemacht werden, so ist er in die Datei `$HOME/.bash_aliases` einzutragen.
- Dazu kann jeder Texteditor verwendet werden, z.B. `nano`.
- Danach ist die Shell zu beenden mit `exit` oder `Ctrl+D`.
- Wird ein neues Terminal geöffnet und damit eine neue Shell gestartet, dann stehen alle in `$HOME/.bash_aliases` eingetragenen Aliase zur Verfügung.

# Man-Pages

- Meistens werden zu einem Kommando auch die zugehörigen Man-Pages installiert.
- Der Befehl `man` gefolgt von einem Kommando öffnet die Man-Page des Kommandos.
- Mit der Return-Taste kann man in einer Man-Page zeilenweise weiterblättern.
- Mit der Leertaste kann man in einer Man-Page seitenweise weiterblättern.
- Mit der q-Taste kann man die Anzeige einer Man-Page beenden.

- Mit der h-Taste findet man die Hilfe für die Navigation in der Man-Page.

```
hermann@debian:~$ man cowsay # show man page of cowsay command
```

```
COWSAY(6)                                Games Manual                                COWSAY(6)

NAME
    cowsay/cowthink - configurable speaking/thinking cow (and a bit more)

SYNOPSIS
    cowsay [-e eye_string] [-f cowfile] [-h] [-l] [-n] [-T tongue_string] [-W column] [-bdgpstwy]

DESCRIPTION
    Cowsay generates an ASCII picture of a cow saying ...
    ...
```

```
hermann@debian:~$ man date # show man page of date command
```

```
DATE(1)                Dienstprogramme für Benutzer                DATE(1)

BEZEICHNUNG
    date - Ausgeben oder Setzen von Systemdatum und -zeit

ÜBERSICHT
    date [OPTION]... [+FORMAT]
    date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]

BESCHREIBUNG
    Die aktuelle Uhrzeit im angegebenen FORMAT anzeigen oder die Systemzeit setzen.

    . . .
```

```
hermann@debian:~$ man man # show man page of man command
```

```
MAN(1)                Dienstprogramme für Handbuchseiten      MAN(1)
```

#### BEZEICHNUNG

man - eine Oberfläche für die System-Referenzhandbücher

#### ÜBERSICHT

```
man [man Optionen] [[Abschnitt] Seite ...] ...
man -k [apropos Optionen] Regulärer_Ausdruck ...
man -K [man Optionen] [Abschnitt] Begriff ...
man -f [whatis Optionen] Seite ...
man -l [man Optionen] Datei ...
man -w|-W [man Optionen] Seite ...
```

#### BESCHREIBUNG

man ist das System-Anzeigeprogramm für die Handbuchseiten. ...

Die folgende Tabelle zeigt die Nummern der Abschnitte der Handbuchseiten und den Typ der dort zu findenden Seiten.

- |   |  |
|---|--|
| 1 | Ausführbare Programme oder Shell-Befehle                     |
| 2 | Systemaufrufe (Kernel-Funktionen)                            |
| 3 | Bibliotheksaufrufe (Funktionen in Programmbibliotheken)      |
| 4 | Spezielle Dateien (gewöhnlich in /dev)                       |
| 5 | Dateiformate und Konventionen, z.B. /etc/passwd              |
| 6 | Spiele   |
| 7 | Verschiedenes (...), z. B. man(7), groff(7), man-pages(7)    |
| 8 | Befehle für die Systemverwaltung (in der Regel nur für root) |
| 9 | Kernel-Routinen [nicht Standard]                             |

Eine Handbuchseite besteht aus verschiedenen Abschnitten.

# Shutdown und Reboot

Auch zum Herunterfahren und Neustarten des Systems ist der `sudo`-Befehl erforderlich.

```
hermann@debian:~$ sudo shutdown now # shutdown the system immediately
```

oder

```
hermann@debian:~$ sudo reboot now # reboot the system immediately
```