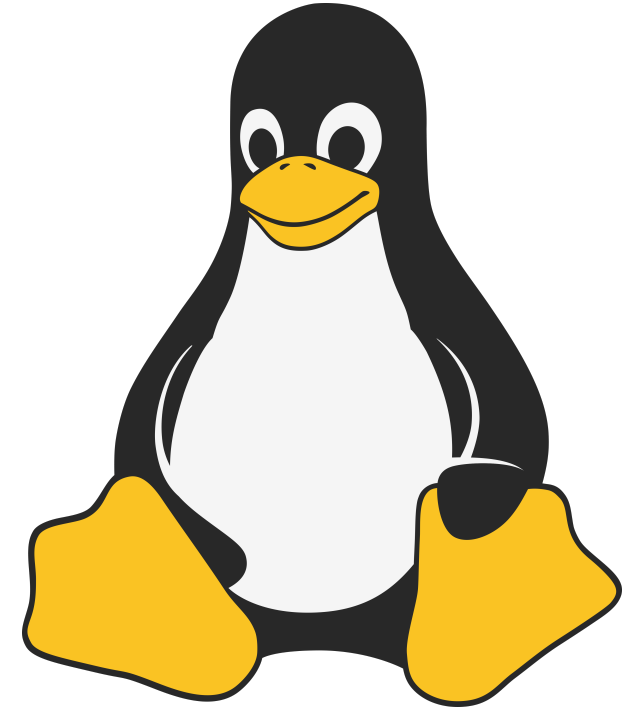


# Shell-Praxis (Teil 1)



Shell-Basics, Variablen-Substitution, **sudo**  
und Paket-Verwaltung mit **apt**

# Inhaltsverzeichnis

- [Vorbemerkungen](#)
- [Die Bash](#)
- [Der `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`
- Aufgaben

# Vorbemerkungen

Die folgenden Beispiel-Sitzungen wollen anhand von Beispielen die Arbeit mit der Shell (Bash) im Terminal (oder einem Terminal-Emulator) vorstellen. Der Schwerpunkt liegt auf der praktischen Anwendung der Shell und der wichtigsten Kommandos. Erklärungen und Erläuterungen sind auf das notwendige Minimum beschränkt.

# Die Bash

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). Sie ist eine Weiterentwicklung der Bourne-Shell (sh) und bietet viele Erweiterungen und Verbesserungen.

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>
```

## `sudo -i`

- Mit `sudo -i` wird eine interaktive `sudo`-Session gestartet.
- Danach kann man mehrere mit root-Berechtigung ausführen, ohne jedes Mal das Passwort eingeben zu müssen.
- Mit `exit` oder `Ctrl+D` wird die interaktive `sudo`-Session beendet. Danach kehren Sie zur normalen Benutzer-Shell zurück.

# 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.

# Aufgaben

Spielen Sie die vorangegangene Sitzung auf Ihrem eigenen Linux-System nach.

- Aktuelles Verzeichnis ermitteln mit `pwd`
- Umgebungsvariablen `$USER`, `$HOME` und `~` anzeigen
- System aktualisieren mit `sudo` und `apt`

- Paket `nca1` installieren, deinstallieren und wieder installieren
- Datum und Kalender ausgeben
- Befehlshistorie anzeigen und Befehle wiederholen
- `ping`-Befehl ausführen (z.B. `ping google.com`) und mit `Ctrl+C` abbrechen