



SV3: Linux Server

Die Shell Teil 1

Agenda

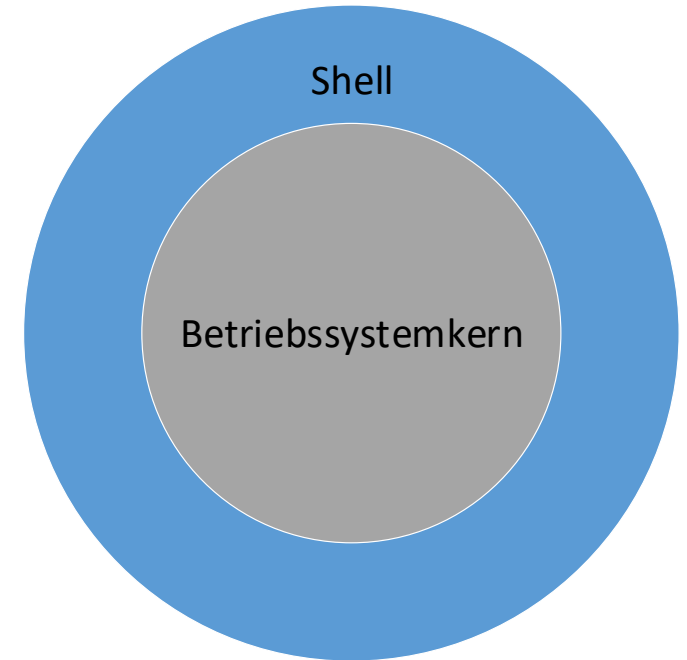
Die Shell

- Grundlagen der Shell
- Funktionen einer Shell
- Grundlegende Kommandos

Was ist eine Shell?

»Shell« hat nichts mit schellen oder klingeln zu tun, sondern Shell kommt aus dem Englischen und bedeutet Muschel oder Schale.

- Sie legt sich wie eine Schale um den Betriebssystemkern und ermöglicht dem Benutzer, im Dialog seine Anweisungen an Linux/Unix zu geben

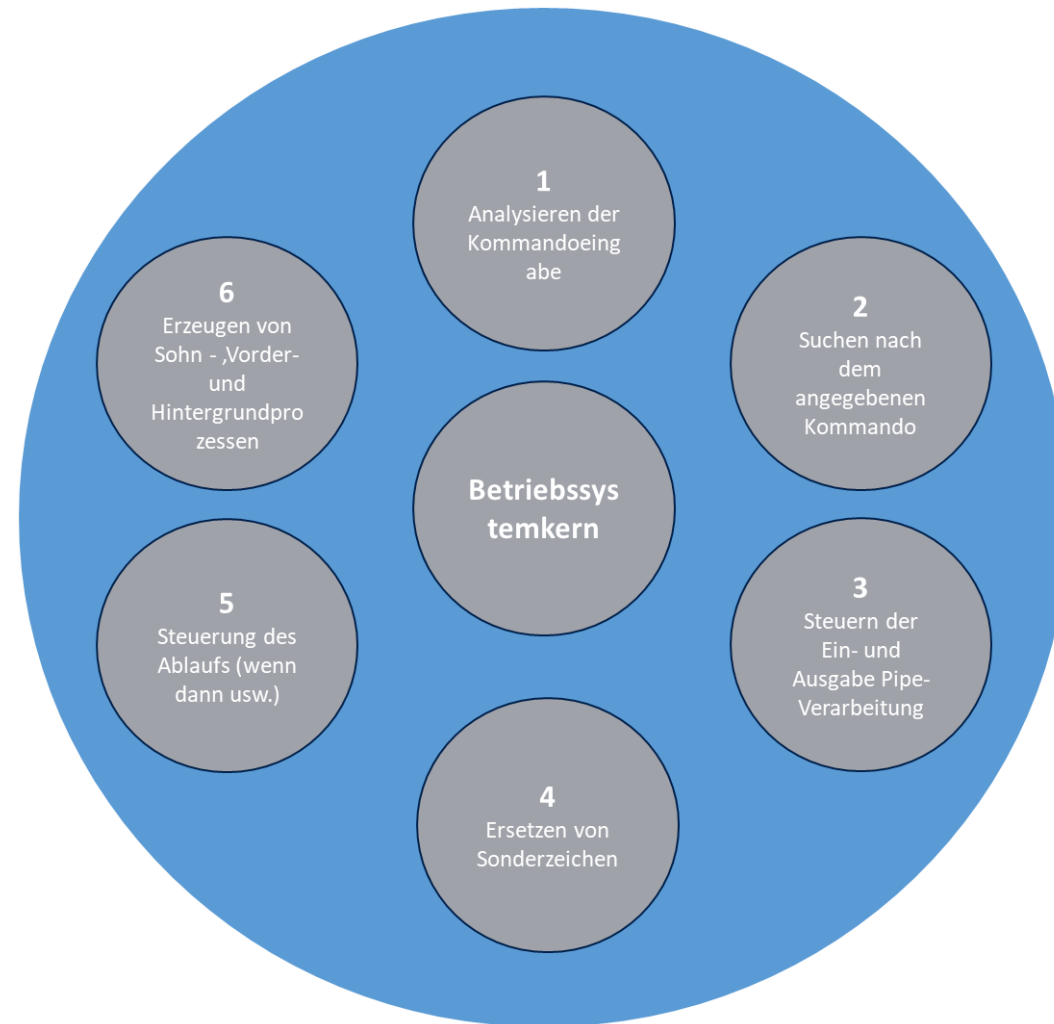


Was ist eine Shell?

Eine Shell ist quasi das Programm, in das Sie Ihre Befehle und Kommandos zum Aufruf von Programmen eingeben.

- Möchten Sie beispielsweise das Programm X starten, so erteilen Sie der Shell den Befehl dazu.
- Die Shell interpretiert das Kommando und leitet es bei Bedarf an den Kernel weiter (z.B., um ein Programm auszuführen).

Grundlagen der Shell - Die Aufgaben der Shell



Login-Shell

Nach dem erfolgreichen Login wird die Login-Shell eines Benutzers gestartet. Von dieser Shell aus werden alle Programme der Arbeitssession gestartet, aber auch Subshells als Kind-Prozesse erzeugt.

- Beim Verlassen der Login-Shell logt man sich aus dem System aus. Zur Weiterarbeit ist eine erneute Anmeldung über das login-Programm notwendig.

Login-Shell versus NoLogin-Shell

Unix unterscheidet beim Aufruf einer Shell, ob es sich um eine Login-Shell handelt, oder um eine No-Login-Shell. Der Unterschied ist einfach:

- Die Login-Shell ist die Shell, die beim Login vom login-Programm gestartet wurde. Jede weitere Shell, die von der Login-Shell aufgerufen wird ist eine No-Login-Shell.
- Der Grund für diese Unterscheidung liegt in der Frage der Konfigurationsdateien, die wir später noch genauer darstellen werden.
- Die Login-Shell muss konfiguriert werden, sie erhält beim Start viele Variablen, die sie an spätere Shells weitervererbt.

Grundlagen der Shell

Kommandos in der Shell

Die Shell nimmt das 1. Wort der eingegebenen Zeile als Kommando und sucht das dazugehörige Programm in vorgegebenen »Suchpfaden«.

- Dem Programmaufruf übergibt die Shell den Rest der Zeile als Anweisung.
- Alle nicht mit - beginnenden Parameter werden von ls als Dateiname oder Directory betrachtet.

Was wurde eingegeben?

Die Shell analysiert die Eingabe wie folgt:

ls

-l

../sebastian

1. Wort = Kommando

Option

Parameter

Grundlagen der Shell

Wo findet nun die Shell das ls-Programm? Gibt es das Kommando überhaupt?

Die Shell sucht in bestimmten Directories nach »ausführbaren« Dateien mit dem angegebenen Namen.

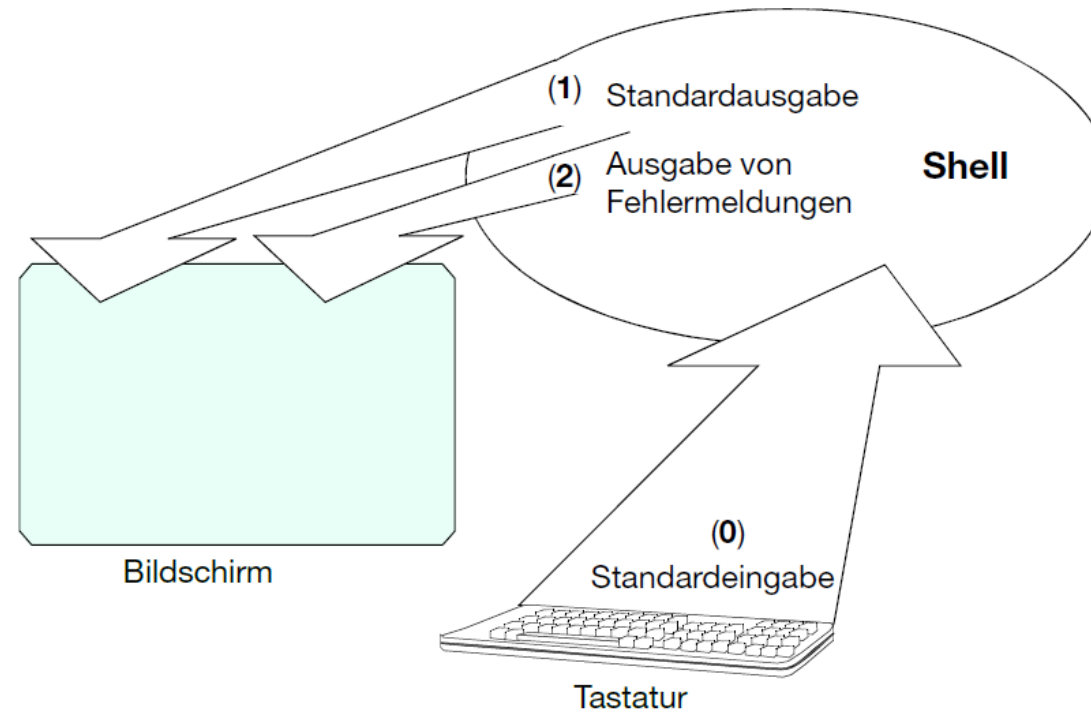
- Für jeden Benutzer ist ein PATH (ein Suchpfad) eingerichtet, der die Namen der Directories enthält.
- Die zumeist unter PATH vorgegebenen Directories sind:
 - /bin /usr/bin
 - Unter diesen Directories sind die Kommandos für den »normalen« Benutzer abgelegt.

Das Linux/Unix-System ist als Dialogsystem konzipiert

d.h., jeder Benutzer kann direkt von seinem Arbeitsplatz (dem Terminal) Kommandos aufrufen, und das Ergebnis wird in der Regel auf seinem Bildschirm angezeigt.

- Die Standardein- und -ausgabe ist das Benutzerterminal
- Durch einfache Steuerzeichen in der Shell kann die Ein- und Ausgabe umgeleitet werden. So bedeutet das Größerzeichen > »Umleitung nach«, das Kleinerzeichen < »Umleitung von«. Dazu aber später mehr.

Standardein- und Standardausgabe der Shell



Grundlagen der Shell

- Sobald die Shell Ihre Aufgaben ordnungsgemäß erfüllt hat, meldet sie sich sofort wieder an Ihrem Bildschirm, dass sie bereit ist, weitere Aufträge entgegenzunehmen.
- Auf dem Bildschirm erscheint dann ein Bereitzeichen (auch Prompt genannt), abhängig vom Programm und Benutzer

Zeichen	Bedeutung
\$	Bereitzeichen für den normalen Benutzer (unter der Bash name@rechner:aktuelles Directory)
%	In der C-Shell
#	Bereitzeichen für den Systemverwalter
>	Kennzeichen, dass die Shell weitere Eingaben für einen Kommandoaufruf erwartet

Grundlagen der Shell

Einige Shell-Programme

Name der Shell	Aufruf	Kurze Beschreibung	Charakteristisch
Korn-Shell in etwas gleich mit Posix-Shell	/bin/ksh Oft auch /bin/sh	Kompatibel zur Bourne-Shell mit wesentlichen Ergänzungen u.a. Historie-Mechanismus, Alias, Befehlszeilen-Editor	Komfortabel und mächtig
Bash (born again shell)	/bin/bash	Unter Linux eingesetzte frei verfügbare Shell (GNU) wird mehr auch unter Unix-Derivaten verwendet.	Noch komfortabler als die Posix-Shell
C-Shell	/bin/csh Oder /bin/tcsh	Ähnlichkeit zur Programmiersprache C – ebenso Historie, Alias, Editor – aber nicht kompatibel zur Bash	Wird gerne von Entwicklern genutzt

Bourne Again Shell

ist ein englisches Wortspiel: Die bash ist somit die wiedergeborene Bourne-Shell, die neben der Korn-Shell und der C-Shell zu den drei klassischen Unix-Shells zählt.

- Unter Linux sind alle drei Shells und noch einige weitere verfügbar, standardmäßig wird aber zumeist die bash eingerichtet

Grundlagen der Shell

Linux auf einem Blatt - Hier sind die wichtigsten Befehle zusammen gefasst.

Grundlegende Befehle, nur ein paar..

- 1. **clear**: Eingabe-Fenster aufräumen
- 2. **whoami**: - wer bin ich - und Who
- 3. **pwd**: Wo bin ich?
- 4. **df**: Dateisystem und Speicherplatz anzeigen
- 5. **passwd**: Passwort ändern
- 6. **ls**: Verzeichnis anzeigen
- 7. **cd**: Verzeichnis wechseln
- 8. **cp**: Kopieren und umbenennen
- 9. **rm**: Dateien löschen
- 10. **cat**: Schnell eine Textdatei anzeigen oder erstellen
- 11. **echo**: Text ausgeben
- 12. **mkdir /rmdir**: Ordner erstellen oder löschen

Linux auf einem Blatt

Christian Helmbold – 2012-05-02
<http://helmbold.de/artikel/linux-blatt>

Hilfe

Klassische Hilfe: **man** (verlassen mit Q)
Online Hilfe von GNU: **info** (verlassen mit Q)
Schlüsselwörter in man-Seiten suchen: **apropos**
Kurzbeschreibung zu einem Kommando oder Schlüsselwort anzeigen: **whatis**

Dateien

Ausgeben von Dateien

Dateien nacheinander ausgeben: **cat**
Datei seitenweise ausgeben: **more**, **less**
Anfangszeilen einer Datei ausgeben: **head**
Datei ab bestimmter Zeile ausgeben: **tail**
Datei mit Zeilennummern ausgeben: **nl**
Datei oktal/hexadezimal ausgeben: **od**, **hd**

Auflisten und Analysieren von Dateien

Dateien auflisten: **ls**
Dateiattribute auflisten: **lsattr**
Dateityp ermitteln: **file**
Programmdatei von Befehl ermitteln: **type**
Zahlen von Zeichen, Wörtern und Zeilen: **wc**
Prüfsumme für eine Datei ermitteln: **sum**, **md5sum**, **sha1sum**
Anzeigen, welcher Prozess eine Datei oder einen Socket geöffnet hat: **fuser**
Geöffnete Dateien anzeigen: **lsof**

Kopieren, Umbenennen und Löschen von Dateien

Kopieren von Dateien: **cp**
Umbenennen/verschieben von Dateien: **mv**
Link auf Datei setzen: **ln**
Dateien oder Verzeichnisbäume löschen: **rm**

Suchen

Suchen nach Dateien: **find**
Schnelles Suchen nach Dateien: **locate**, **slocate**

Vergleichen von zwei Dateien: **cmp**
Zusammenführen von Dateien: **join**
Sortieren von Dateien: **sort**

Komprimieren und Archivieren von Dateien

(De-)Komprimieren von Dateien: **gzip/gunzip**, **pack/unpack**, **compress/uncompress**, **arc**, **unrar**, **unrar**
(De-)Komprimieren von Dateien mit bzip: **bzip2**, **bunzip2**, **bzcat**
Inhalt von gzip-Archiven anzeigen: **zcat**, **zless**, **zmore**

Archivieren bzw. Kopieren von Dateien und Dateibäumen: **tar**, **cpio**

Ändern von Zugriffsrechten, Eigentümer und Zeitstempel

Zugriffsrechte einer Datei ändern: **chmod**
Eigentümer einer Datei ändern: **chown**
Gruppe einer Datei ändern: **chgrp**
Attribute einer Datei ändern: **chattr**
Dateikreierungsmaske setzen bzw. ausgeben: **umask**

Ändern des Zeitstempels einer Datei: **touch**

Umformen, Extrahieren und Zerteilen von Dateien

Zeichen in Dateien ersetzen: **tr**
Zeichenketten transformieren: **sed**
Tabulatoren in Leerzeichen umwandeln: **expand**
Zeichensätze konvertieren: **recode**, **iconv**
Identische, aufeinander folgende Zeilen nur einmal ausgeben: **uniq**
Herausschneiden von Spalten oder Feldern aus Dateien: **cut**

Zerteilen von Dateien: **split**, **esplit**

Drucken

Datei auf Drucker ausgeben: **lp**, **lpr**
Statusinformationen zu Druckaufträgen erfragen: **lpstat**, **lpq**
Druckaufträge abbrechen: **cancel**
Drucker verwalten: **lpc**
Druckaufträge löschen: **lprm**

Vergleichen zweier Verzeichnisse: **diff**
Basiname eines Pfades: **basename**
Verzeichnisname eines Pfades: **dirname**

Speicherplatzinformationen

Freien Speicherplatz ermitteln: **df**
Speicherbedarf von Dateien oder Verzeichnissen ermitteln: **du**
Anzeigen des freien Hauptspeichers und Swap: **free**

Dateisysteme

Einhängen, Partitionieren, Formatieren und Kopieren

Einhängen eines Dateisystems: **mount**
Aushängen eines Dateisystems: **umount**
Partitionieren einer Festplatte: **fdisk**, **cfdisk**
Anlegen, verkleinern, vergrößern und verschieben von Partitionen: **parted**
Disketten formatieren: **fdformat**
Kopieren und konvertieren von Dateisystemen und Partitionen: **dd**

Einrichten und Prüfen von Dateisystemen

Einrichten von Dateisystemen: **mkfs**
Anlegen eines ext2- oder ext3-Dateisystems: **mke2fs**
ReiserFS-Dateisystem anlegen: **mkreiserfs**
Swap-Partitionen und -Dateien einrichten: **mkswap**, **swapon**, **swapoff**
Prüfen und Reparieren eines Dateisystems: **fsck**
Physikalische Prüfung eines Datenträgers: **badblocks**

Weitere Dateisystembefehle

Informationen zu einem ext2-/ext3-Dateisystem: **dumpe2fs**
Systemparameter eines ext2-/ext3-Dateisystems ändern: **tune2fs**
Gepufferte Daten auf die Festplatte schreiben: **sync**
Zugriff auf MS-DOS-Disketten: **mttools**

Benutzer und Gruppen

Benutzer anlegen: **useradd**, **adduser**

Grundlagen der Shell

Datumsanzeige

Unter Linux/Unix ist es wichtig, mit dem richtigen Datum zu arbeiten

- u.a. für Sicherungen und für Updates bei Programmdateien.

Wie spät ist es?

`date [+ „Formatangaben“]`

Datum

Änderung der Ausgabe

```
sb@ubuntu01: ~  
Datei Bearbeiten Ansicht Suchen Terminal Hilfe  
sb@ubuntu01:~$ date  
Mi 27. Jun 11:18:51 CEST 2018  
sb@ubuntu01:~$ date +"heute ist der: %d. %m."  
heute ist der: 27. 06.  
sb@ubuntu01:~$
```


Grundlagen der Shell

Lesen des Online-Manuals

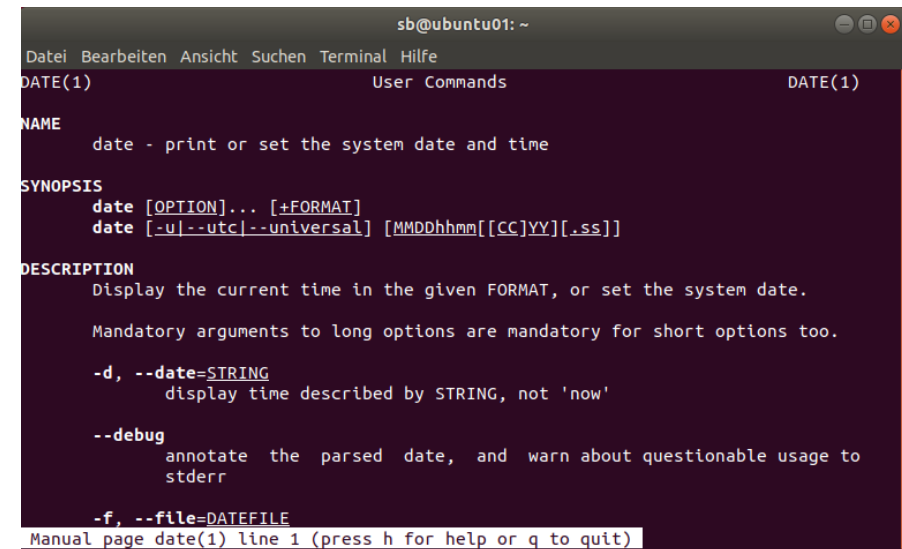
Unter dem Directory `/usr/man` sind nach Kapiteln getrennt (`man1` - `man8`) die Beschreibungen (Manualeseiten) der Kommandos abgelegt.

- Mit dem Kommando `man` können Sie sich diese Seiten am Bildschirm ansehen

man Kommando

manual

man – Kommando, um die Beschreibung von Kommandos (Manualeseiten) anzusehen



```
sb@ubuntu01: ~  
Datei Bearbeiten Ansicht Suchen Terminal Hilfe  
DATE(1) User Commands DATE(1)  
  
NAME  
    date - print or set the system date and time  
  
SYNOPSIS  
    date [OPTION]... [+FORMAT]  
    date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]  
  
DESCRIPTION  
    Display the current time in the given FORMAT, or set the system date.  
  
    Mandatory arguments to long options are mandatory for short options too.  
  
    -d, --date=STRING  
        display time described by STRING, not 'now'  
  
    --debug  
        annotate the parsed date, and warn about questionable usage to  
        stderr  
  
    -f, --file=DATEFILE  
        Manual page date(1) line 1 (press h for help or q to quit)
```

Unter Linux gibt es zu einigen Kommandos auch weitere Hilfeabfragen.

Kommando **--help**

--help – gibt unter Linux eine Kurzinformation über einige Kommando

```
root@SBNKFAP01:~# date --help
Aufruf: date [OPTION]... [+FORMAT]
oder: date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]
Anzeige der aktuellen Zeit im angegebenen FORMAT oder Setzen der Systemzeit.

Erforderliche Argumente für lange Optionen sind auch für kurze erforderlich.
-d, --date=ZEICHENKETTE  Zeit gemäß ZEICHENKETTE anzeigen, nicht „jetzt“
-f, --file=DATEI         wie --date für jede Zeile in DATEI
-I[ZEITSPEZ], --iso-8601[=ZEITSPEZ]  gibt Datum/Zeit im ISO 8601 Format aus.
                                ZEITSPEZ='date' nur für Datum (Vorgabe),
```

Grundlagen der Shell

Wenn Sie nur wissen wollen, für was ein bestimmtes Kommando genutzt werden kann, geben Sie Folgendes ein:

whatis Kommando

whatis – Kommando, um eine Kurzinformation über ein Kommando zu erhalten

```
root@SBNKFAP01:~# whatis ls
ls (1)                - list directory contents
root@SBNKFAP01:~#
```

Grundlagen der Shell

Anzeige eines Dateiinhaltes

Um zu sehen, welchen Inhalt eine Datei hat, gibt es das Kommando cat.

cat Dateiname

concatenate (zusammenfügen)

Cat – Kommando, um sich einen Dateiinhalt anzusehen.

```
sb@ubuntu01:~$ cat .bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
    *) ;;
    *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000
```

Grundlagen der Shell

Blättern im Inhalt einer Datei

Um einen Dateiinhalt seitenweise am Bildschirm anzusehen, gibt es zwei sehr ähnliche Kommandos:

less Dateiname(n)

less (less is more than more)

less – Kommando, um Dateiinhalte seitenweise anzuzeigen.

more Dateiname(n)

more (mehr)

more – Kommando, um Dateiinhalte seitenweise anzuzeigen.

Leertaste	Seitenweise vorwärts blättern
Eingabetaste	Zeilenweise vorwärts gehen
h	Aufruf der Hilfe
q	Beenden des Programms (quit) Bei Dateiende wird das Programm automatisch beendet.

```
sb@ubuntu01: ~  
Datei Bearbeiten Ansicht Suchen Terminal Hilfe  
# System-wide .bashrc file for interactive bash(1) shells.  
  
# To enable the settings / commands in this file for login  
# shells as well,  
# this file has to be sourced in /etc/profile.  
  
# If not running interactively, don't do anything  
[ -z "$PS1" ] && return  
  
# check the window size after each command and, if necessary,  
# update the values of LINES and COLUMNS.  
shopt -s checkwinsize  
  
# set variable identifying the chroot you work in (used in  
# the prompt below)  
/etc/bash.bashrc
```

Grundlagen der Shell

Wer arbeitet am System?

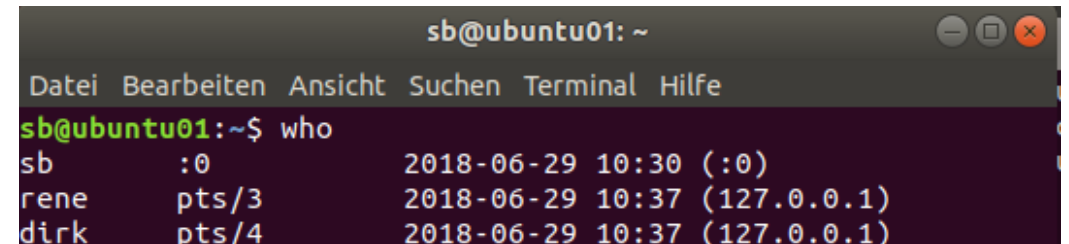
Mit dem Kommando `who` erfahren Sie, wer außer Ihnen noch am System arbeitet.

- Sie sehen, mit welchem Namen derjenige im System angemeldet ist, an welchem Terminal er arbeitet und um wieviel Uhr er sich angemeldet hat.

who

(wer)

`who` – Kommando, um zu sehen, wer noch am System arbeitet



```
sb@ubuntu01: ~  
Datei Bearbeiten Ansicht Suchen Terminal Hilfe  
sb@ubuntu01:~$ who  
sb      :0      2018-06-29 10:30 (:0)  
rene    pts/3   2018-06-29 10:37 (127.0.0.1)  
dirk    pts/4   2018-06-29 10:37 (127.0.0.1)
```

Benutzername Name des Terminals Datum und Uhrzeit der Anmeldung (Login)

Grundlagen der Shell

Ihre Benutzerkennung (Namen, mit dem Sie sich angemeldet haben), den Namen des Terminals, an dem Sie arbeiten, und das Datum und die Uhrzeit Ihrer Anmeldung

whoami

Wer bin ich?

```
sb@ubuntu01: ~  
Datei Bearbeiten Ansicht Suchen Terminal Hilfe  
sb@ubuntu01:~$ whoami  
sb  
sb@ubuntu01:~$
```

Grundlagen der Shell

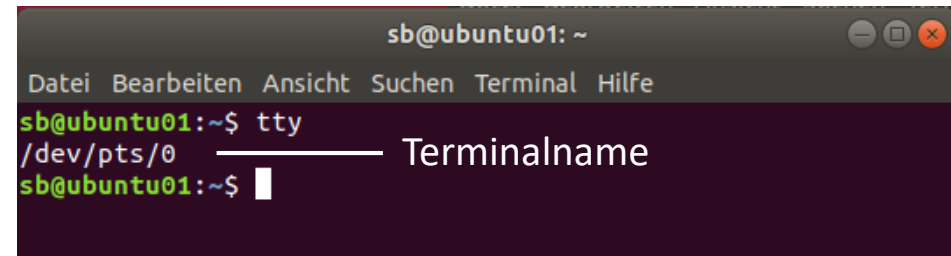
Anzeige des Terminalnamens

Sollten Sie Schwierigkeiten mit Ihrem Terminal haben und können Sie plötzlich keine weiteren Eingaben mehr machen oder erhalten kein Login, um sich anzumelden, dann ist es für den Systemverwalter von Vorteil, wenn er den Namen Ihres Terminals weiß.

tty

terminal type (Terminaltyp)

tty – Kommando, um den Terminalnamen (Typ) anzuzeigen.



```
sb@ubuntu01: ~  
Datei Bearbeiten Ansicht Suchen Terminal Hilfe  
sb@ubuntu01:~$ tty  
/dev/pts/0      Terminalname  
sb@ubuntu01:~$
```

Arbeiten Sie mit einer grafischen Oberfläche, wird jedem geöffneten Fenster (Terminal) ein eigener Name zugewiesen. Meist wird es als Pseudoterminal mit `/dev/pts/n` bezeichnet (*n* steht für die Nummer, 1 für das erste geöffnete Fensters, 2 für das zweite usw.).

Suchen unter Linux mit find

Sucht man z. B. eine bestimmte Datei, von der genaue Dateiname noch bekannt ist, ist das folgende Beispiel einen Versuch wert:

```
find . -name tux.jpg
```

- Bei diesem Beispiel wird die Datei tux.jpg gesucht. Der Punkt bewirkt das alle Verzeichnisse ab den aktuellen durchsucht werden.

Suchen unter Linux mit find

Man kann die Suche natürlich ein wenig steuern, indem man das vermutete Verzeichnis direkt angibt:

```
find /home/dirk/ -name tux.jpg
```

- oder man entscheidet sich für die ungeschnittene Version:

```
find / -name tux.jpg
```

- Wobei durch den `/` angegeben wird, dass find im Stammverzeichnis anfängt.

Wie auch bei **locate** muss man sich nicht wundern, dass man lauter Fehlermeldungen erhält, wenn man nicht als root angemeldet ist. Da man nicht auf jedes Verzeichnis zugreifen darf.

Grundlagen der Shell

Wichtig!

Bei den oben angegebenen Beispielen ist die Suche Case-Sensitive, was bedeutet, dass find Groß- und Kleinschreibung unterscheidet. Möchte man dies unterbinden, kann man das mit dem Argument **-iname** verhindern.

■ Also beispielsweise:

```
find / -iname tux.jpg
```

Texteditoren unter Linux

- ed
- vi / vim
- nano
- weitere..

Der vi gehört nach wie vor zu den mächtigsten Editoren

die unter Linux/Unix verfügbar sind. Doch der vi hat den Vorteil, dass er auf jedem Unix-/Linuxrechner vorhanden ist und mit und ohne grafische Oberfläche funktioniert.

- Die Cursortasten dürfen allerdings nicht im Eingabemodus verwendet werden (Ausnahme: der vi unter Linux »vim«. Hier sind die Cursortasten auch im Eingabemodus erlaubt).
- den Kommandomodus, den Eingabemodus einen ed-ähnlichen Modus (ex-Modus).

Grundlagen der Shell

vi

Um Text einzufügen, müssen Sie erst durch ein entsprechendes Kommando in den Eingabemodus umschalten. Die Texteingabe kann eingeschaltet werden durch folgende Kommandos:

Kommando	Ableitung von	Bedeutung
A, a	append	Anhängen ans Zeilenende bzw. hinter dem Cursor
I, i	Insert	Einfügen am Zeilenanfang bzw. vor dem Cursor
R	replace	Ersetzen ab Cursor
O, o	Open	Neue Zeile über bzw. unter der aktuellen Zeile einfügen.

Der Eingabemodus wird durch die Funktionstaste Escape abgeschlossen.

Grundlagen der Shell

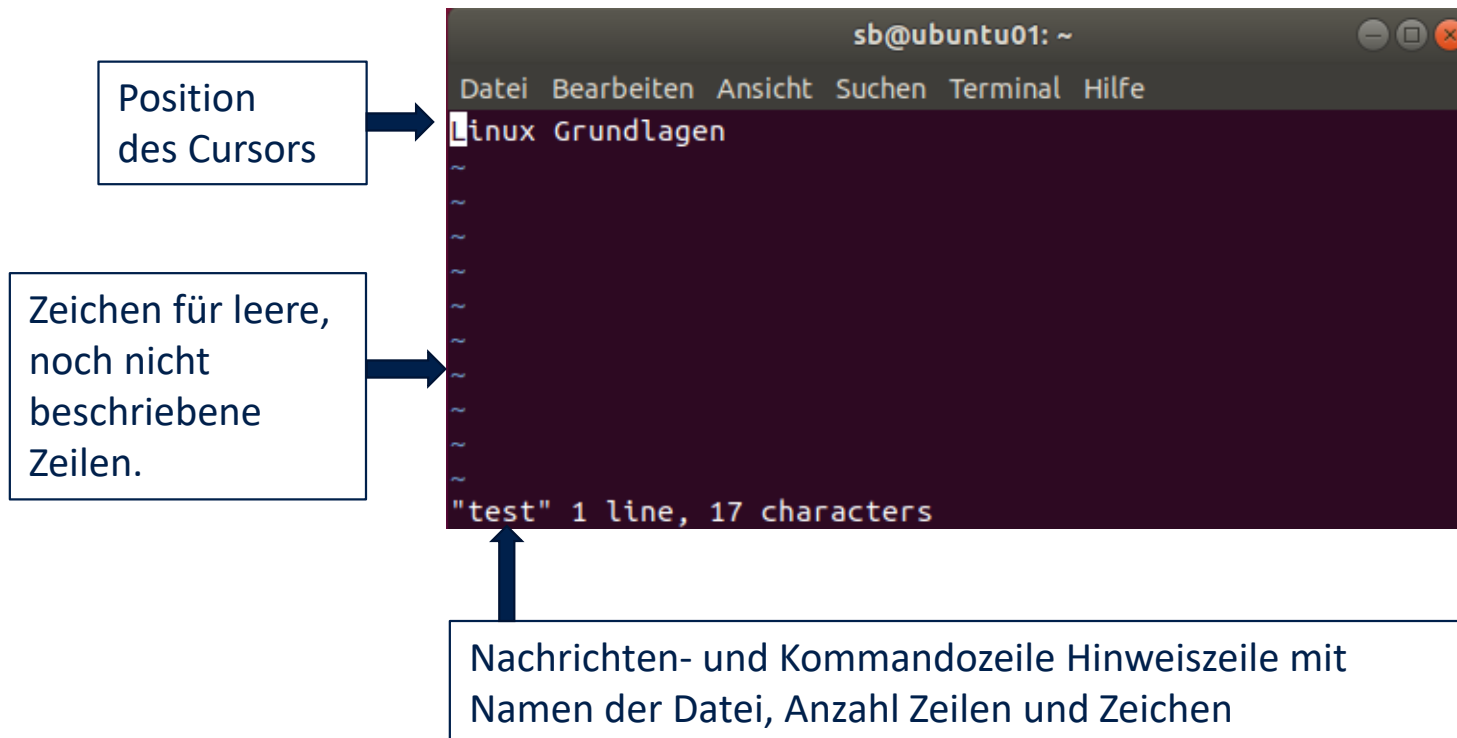
vi

Alle erfolgten Änderungen müssen durch ein Sicherungskommando in die Originaldatei zurückgeschrieben werden. Die Kommandos zum Sichern (nur im Befehlsmodus) und Beenden des vi sind unten aufgeführt.

Kommando	Ableitung von	Bedeutung / Funktion
:w	write	Rückschreiben des Arbeitspuffers auf die Platte.
:q	quit	Beenden des vi, mit Warnung falls noch nicht geschrieben wurde.
:q!	quit	Beenden des vi ohne Warnung.
:wq	write & quit	Zurückschreiben des Arbeitspuffers und Beenden des vi.

Grundlagen der Shell

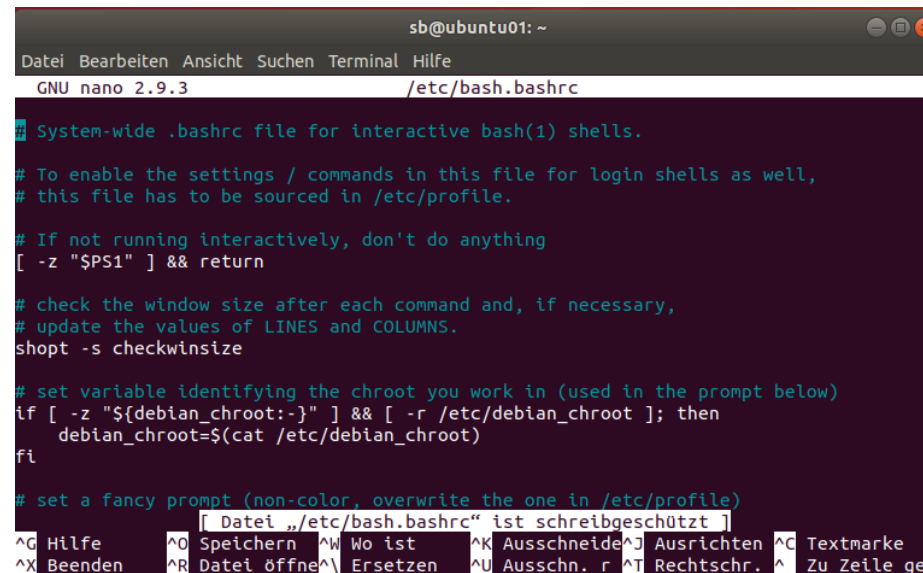
Vi



Grundlagen der Shell

Nano

Zwar bescheiden im Befehlsumfang, aber dafür einfach zu bedienen, ist nano bzw. pico. Bei den meisten aktuellen Distributionen ist lediglich nano installiert.



The screenshot shows a terminal window with the nano text editor open, editing the file /etc/bash.bashrc. The window title is 'sb@ubuntu01: ~'. The nano menu bar at the top includes 'Datei', 'Bearbeiten', 'Ansicht', 'Suchen', 'Terminal', and 'Hilfe'. The status bar at the bottom shows the current file path and various keyboard shortcuts. The editor content includes comments and code for setting up the bash environment, such as sourcing /etc/profile, checking for interactive shells, and setting a prompt.

```
sb@ubuntu01: ~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
GNU nano 2.9.3 /etc/bash.bashrc

# System-wide .bashrc file for interactive bash(1) shells.

# To enable the settings / commands in this file for login shells as well,
# this file has to be sourced in /etc/profile.

# If not running interactively, don't do anything
[ -z "$PS1" ] && return

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize

# set variable identifying the chroot you work in (used in the prompt below)
if [ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ]; then
    debian_chroot=$(cat /etc/debian_chroot)
fi

# set a fancy prompt (non-color, overwrite the one in /etc/profile)
[ Datei „/etc/bash.bashrc“ ist schreibgeschützt ]
^G Hilfe      ^O Speichern  ^W Wo ist     ^K Ausschneide^J Ausrichten  ^C Textmarke
^X Beenden    ^R Datei öffne^_ Ersetzen   ^U Ausschn. r ^T Rechtschr. ^_ Zu Zeile ge
```

Grundlagen der Shell

Nano

Im unteren Bereich befindet sich eine kleine Übersicht über mögliche Aktionen:

```
^G Hilfe      ^O Speichern ^R Datei öffn ^Y Seite zurü ^K Ausschneid ^C Cursor  
^X Beenden    ^J Ausrichten ^W Wo ist     ^V Seite vor  ^U Ausschn. r ^I Rechtschr.
```

Die ^ steht jeweils für [STRG]+ den folgenden Buchstaben - Beispiele:

- Mit [STRG]+O kann man die Datei abspeichern
- mit [STRG]+X kann man das Editorfenster schließen
- eine Suche erreicht man über [STRG]+W
- [STRG]+C zeigt die aktuelle Position des Cursors an (Zeile, Spalte)
- usw..



**WBS
TRAINING**

**VIELEN DANK
FÜR IHRE
AUFMERKSAMKEIT!**



UNSER  SCHLÄGT FÜR BILDUNG