

Die Linuxkonsole

Alles was man für das schnelle Arbeiten braucht :-)

O. Krisch

Linux User Gruppe Schwabach

2015 / SLT



Warum verwendet man die Konsole und nicht die graphische Oberfläche?

Ein kurzer Erfahrungsbericht

- Nicht alle Optionen für die Befehle sind graphisch umgesetzt.
- Manches ist schneller getippt als geklickt.
- Wiederkehrende Aufgaben können in ein Skript geschrieben werden.

Was kann ich sonst noch damit machen ?

- Briefe oder Texte schreiben. (Vim, Emacs, Nano)
- E-Mails verschicken und empfangen (Emacs)
- Surfen
- Den Computer administrieren. (Konsole)
- ...

Der Tag, an dem wir Kontakt aufnehmen

```
oliver@Ganymed:~/Dokumente$
```

- Benutzer
- Hostname
- Arbeitsverzeichnis
- „Status“ (# Administrator; \$ normaler Benutzer)
- Cursor

Wie sind Kommandos aufgebaut ?

<Kommando> <Option(en)> <Argument(e)>

- Kommando (Was wird gemacht ?)
- Optionen (Wie wird es gemacht ?)
- Argumente (Womit wird etwas gemacht ?)

Beispiel

```
ls -l /home
```

Wir unterscheiden:

interne Kommandos: Sie werden direkt von der Konsole zur Verfügung gestellt und schnell ausgeführt.

externe Kommandos: kleine Programme, die von der Konsole ausgeführt werden.

kurze Optionen: `-a -l -F` oder `-alF`

lange Optionen: `--all --format=long --classify`

Was zu beachten wäre?

- Linux unterscheidet zwischen Groß- und Kleinschreibung.
Bsp.: `Linux.txt` und `linux.txt`
- Sonderzeichen: `$ & ; () { } [] * ? ! < > " '`

Wo bekomme ich Hilfe? (1)

- Für externe Kommandos gibt es die Optionen: `--help`, `-h` oder `-?` (für Kurzhilfen)
- Für interne Kommandos gibt es den Befehl `help`.
Bsp.: `help type`
- `manpages` (Handbuchseiten)
 - Anzeigen mit dem Befehl `man`
Bsp.: `man less`
Beenden mit `q`
 - `apropos` sucht in den Manpages nach Stichwörtern und gibt den Titel aus.
Bsp.: `apropos less`
 - `whatis` sucht nach dem Namen von Kommandos und gibt eine Kurzbeschreibung aus.
Bsp.: `whatis less`

Wo bekomme ich Hilfe? (2)

- HOWTOs
Keine Beschreibung einzelner Befehle, sondern eine Anleitung zum Lösen bestimmter Probleme.
- `/usr/share/doc` Zentraler Ort für die Ablage von zusätzlicher Dokumentation
- Bei Ihrer lokalen LUG um die Ecke.
- ... und natürlich bei der Suchmaschine Ihres Herzens

Wo bekomme ich Hilfe? (3)

Ergänzung zu den Manpages:

Nr.	Themenbereiche
1	Ausführbare Programme oder Shellbefehle
2	Systemaufrufe (Kernelfunktionen)
3	Bibliotheksaufrufe (Funktionen in System-Bibliotheken)
4	Spezielle Dateien (gewöhnlich in /dev)
5	Dateiformate und Konventionen, z. B. /etc/passwd
6	Spiele
7	Makropakete und Konventionen, z. B. man(7), groff(7)
8	Systemadministrationsbefehle (in der Regel nur für root)
9	Kernelroutinen [Nicht Standard]

Wo bekomme ich Hilfe? (4) Übung

Informieren Sie sich mit Hilfe der Befehle `man`, `apropos`, `whatis` und `info` über folgende Befehle:

- `ls`
- `cp`
- `rm`

Grundbefehle im Umgang mit Dateien (1)

- Anlegen einer (leeren) Datei

- **touch**

Bsp.: `touch datei1 datei2`
`touch /home/vhs/datei.txt`

- Kopieren, Verschieben und Löschen

- **cp** (copy)

Bsp.: `cp datei1 datei2` Kopiert eine Datei in eine andere
`cp datei /home/peter` Kopiert eine Datei in ein Verzeichnis
`cp -R /Verz /home` Kopiert ein Verzeichnis mit Inhalt

- **mv** (move)

Bsp.: `mv datei1 datei2` Benennt eine Datei um.
`mv datei /home/peter` Verschiebt eine Datei in ein Verzeichnis

- **rm** (remove)

Bsp.: `rm datei1 datei2` Löscht zwei Dateien
`rm /home/peter/datei.txt` Löscht eine Datei im Verzeichnis
`rm *.txt` Löscht alle Dateien mit „.txt“
`rm -rf *` **Löscht alles!**

- Legen Sie Dateien `TEST1`, `TEST2` und `TEST3` an.
- Kopieren Sie `TEST1` in die Dateien `TEST4` und `TEST5` um.
- Benennen Sie die Dateien um.
- Löschen Sie die Dateien wieder.

Grundbefehle im Umgang mit Dateien (2)

- Dateiinhalte anzeige:
 - **cat** (concatenate) Inhalt ausgeben oder mehrere Dateien verbinden.
Bsp.: `cat datei1 datei2` Hängt den Inhalt der Datei2 an den Inhalt von Datei1
`cat /proc/cpuinfo` Gibt den Inhalt von cpuinfo aus.
 - **tac** Gibt den Inhalt in umgekehrter Reihenfolge aus.
Bsp.: `tac /proc/cpuinfo`
 - **less** oder **more** Zeigt Dateien seitenweise an. (less ist neuer und besser)
Bsp.: `less /proc/cpuinfo`
 - **head** Zeigt nur den Dateianfang an (ersten 10 Zeilen).
Bsp.: `head /proc/cpuinfo`
`head -n 20 /proc/cpuinfo` Gibt die ersten 20 Zeilen aus.
 - **tail** Zeigt nur das Dateiende an (letzten 10 Zeilen).
Bsp.: `tail /proc/cpuinfo`
`tail -n 20 /proc/cpuinfo` Gibt die letzten 20 Zeilen aus.

Grundbefehle im Umgang mit Dateien (3)

- Den Datentyp bestimmen mit **file**

Bsp.: `file konsole.png`

> `Konsole.png: PNG image, 643 x 221, 8-bit/color RGB,`

> `non-interlaced`

- **stat** zeigt weitere Informationen über eine Datei an.
(Beispiel später)
- **grep** sucht in Dateien.
(Dazu evtl. später mehr.)

Grundbefehle im Umgang mit Dateien (4)

- **ln** (link) Dateien mit Links verknüpfen.

Bsp.: `ln <datei> <lndatei>` Verknüpft die ursprüngliche Datei `<datei>` mit einem weiteren Namen `<lndatei>`

- hard link (harter Link)

Bsp.: `ln datei1 datei2`

- soft link (symbolische Links)

Bsp.: `ln -s datei1 datei2`

Was zu beachten ist!

Symbolische Links und harte Links zeigen beim Löschen unterschiedliches Verhalten nach dem Löschen der ursprünglichen Datei:

symbolischer Link: Der Link zeigt auf eine nicht mehr existierende Datei.

harter Link: Die Datei existiert nur noch unter dem anderen Namen.

Ein Baum statt Laufwerke

```
/ <root>      # Wurzelverzeichnis
|-bin/        # Programme für die Systembenutzung
|-boot/       # Dateien für den Bootloader
|-dev/        # Gerätedateien
|-etc/        # Konfigurationen
|-home/       # Heimatverzeichnisse der Benutzer
|-lib/        # Bibliotheken und Kern-Module
|-media/      # Wechselmedien
|-mnt/        # Andere, z.B. entfernte Medien
|-opt/        # Optionale Software
|-proc/       # Schnittstelle zum Kern
|-root/       # Heimatverzeichnis für den Superuser / Administrator
|-sbin/       # Administrationsprogramme
|-srv/        # Inhalte für Serverprogramme
|-sys/        # Schnittstelle zur Hardware
|-tmp/        # Temporärer Speicher
|-usr/        # Benutzerprogramme und Daten
|  |-bin      # Benutzerprogramme
|  |-share
|  |-'-doc/   # Dokumentation
|-var         # Variable Daten
```


Grundbefehle im Umgang mit Verzeichnissen (1)

- **ls** (list segments) Zeigt den Inhalt eines Verzeichnisses an.

Bsp.: **ls -a** alles wird angezeigt auch versteckte Dateien

ls -l lange detailreiche Anzeige

ls -lh wie oben nur mit menschlich lesbaren Größenangaben

ls -r umgekehrte Ausgabe

ls -R Verzeichnisse werden rekursiv angezeigt.

ls -1 pro Zeile ein Datei

Grundbefehle im Umgang mit Verzeichnissen (2)

Beispielausgabe für `ls -l`

```
-rw-r--r-- 1 oliver oliver 3582 2011-03-06 18:28 Konsole.png
drw-r--r-- 2 oliver oliver 4096 2011-03-08 20:16 VHS
```

Erläuterung

Dateityp + Berechtigung	Hardlinks Verzeichnisanzahl	Besitzer	Gruppe	Größe	Änderungs- datum	Datei- name
----------------------------	--------------------------------	----------	--------	-------	---------------------	----------------

einige Dateitypkennzeichnungen in `ls`

Dateityp	Farbe	Sonderzeichen (<code>ls -F</code>)	Kennung (<code>ls -l</code>)
gewöhnliche Datei	schwarz		-
ausführbare Datei	grün	*	-
Verzeichnis	blau	/	d
Link	cyan	@	l

Grundbefehle im Umgang mit Verzeichnissen (3)

- **pwd** (print working directory) Gibt das aktuelle Verzeichnis aus.
- **cd** (change directory) Zum Wechseln in andere Verzeichnisse.

Bsp.: **cd <Pfad/Verz>** Wechselt in das Verzeichnis <Pfad/Verz>
cd .. Wechselt in das übergeordnete Verzeichnis.
Leerzeichen beachten!!!
cd ~ oder **cd** Wechselt in das HOME-Verzeichnis
cd - Wechselt in das letzte besuchte Verzeichnis

relative und absolute Pfadangabe

absoluter Pfad Beginnt immer mit einem „/“, der für das Wurzelverzeichnis steht.

relativer Pfad Beginnt nie mit einem Schrägstrich. „Relativ“ bezieht sich auf das aktuelle Verzeichnis, in dem man sich befindet. Es können auch die Links „.“ (aktuelles Verzeichnis) und „..“ (übergeordnetes Verzeichnis) verwendet werden.

Grundbefehle im Umgang mit Verzeichnissen (4)

- **mkdir** (make directory) Erstellt Verzeichnisse

Bsp.: `mkdir testdir`

- **rmdir** (remove directory) Löscht leere Verzeichnisse

Bsp.: `rmdir testdir`

- **rm -r** (remove) Löscht Verzeichnisse und deren beinhaltete Dateien

Bsp.: `rm -r testdir`

- Einen Benutzer anlegen mit

`useradd <Optionen> <Benutzername>` (alt) oder besser mit
`adduser <Optionen> <Benutzername>`

Optionen sind:

- `-d <Heimatverzeichnis>`
- `-g <primäre Gruppe>`
- `-G <weitere Gruppen>`
- `-s <Loginshell>`
- `-m` Legt das Heimatverzeichnis an
- Das Passwort eingeben bzw. ändern.
`passwd <Benutzer>` Ändert als Root das Passwort des
<Benutzers>. Ohne Angabe eines Benutzernamens kann man sein
eigenes Passwort ändern.
- Benutzerkonten löschen
`userdel <Option> <Benutzername>` (alt) oder besser mit
`deluser <Option> <Benutzername>`
 - `-r` Löscht das Benutzer- und Mailverzeichnis. Der „Rest“ muss per
Hand gelöscht werden.

- Eine Gruppe anlegen mit:

`groupadd <Option> <Gruppenname>` (alt) oder mit
`addgroup <Option> <Gruppenname>`

- `-g <GID>` Angabe einer Gruppenid. Ohne diese Option wird die nächste freie Nummer genommen.

- Eine Gruppe löschen mit:

`groupdel <Gruppenname>` (alt) oder mit
`delgroup <Gruppenname>`

Eigene Mitgliedschaft bestimmen

- `id` gibt die Userid, Gruppenid, Primärgruppe und alle weiteren Gruppen aus.
- `groups` gibt nur die Gruppennamen aus.

Berechtigungen kennen und verstehen (1)

Ausgabe von `ls -l`

insgesamt 1436

```
-rw-r--r-- 1 oliver oliver 1833 2011-03-07 00:05 Konsolenvortrag.aux
-rw-r--r-- 1 oliver oliver 570712 2011-03-07 00:05 Konsolenvortrag.dvi
-rw-r--r-- 1 oliver oliver 41220 2011-03-07 00:05 Konsolenvortrag.log
-rw-r--r-- 1 oliver oliver 988 2011-03-07 00:05 Konsolenvortrag.nav
-rw-r--r-- 1 oliver oliver 0 2011-03-07 00:05 Konsolenvortrag.out
-rw-r--r-- 1 oliver oliver 743486 2011-03-07 00:05 Konsolenvortrag.ps
-rw-r--r-- 1 oliver oliver 0 2011-03-07 00:05 Konsolenvortrag.snm
-rw-r--r-- 1 oliver oliver 14990 2011-03-08 21:21 Konsolenvortrag.tex
-rw-r--r-- 1 oliver oliver 40 2011-03-07 00:05 Konsolenvortrag.toc
-rw-r--r-- 1 oliver oliver 928 2011-03-07 00:05 Konsolenvortrag.vrb
-rw-r--r-- 1 oliver oliver 3582 2011-03-06 18:28 Konsole.png
-rw-r--r-- 1 oliver oliver 52204 2011-03-08 20:51 ls-Ausgabe.png
drwxr-xr-x 2 oliver oliver 4096 2011-03-08 20:49 Test
```

Berechtigungen kennen und verstehen (2)

	Dateityp	Besitzer	Gruppe	Alle
Verzeichnis	d	r w x	r - x	r - x
Datei	-	r w -	r - -	r - -

- **d** Dateityp im Dateisystem
 - d Verzeichnis
 - - normale Datei
 - l Link
- **rwX** Berechtigungen für Besitzer, Gruppe und den Rest der Welt
 - r für Lesen (read)
 - w für Schreiben (write)
 - x für Ausführen bzw. Betreten von Verzeichnissen (execute)

Berechtigungen kennen und verstehen (3)

- stat Zeigt detailliertere Dateiattribute an

```
File: 'Konsole.png'  
Size: 3582          Blocks: 8          IO Block: 4096   reguläre Datei  
Device: 806h/2054d Inode: 2942023   Links: 1  
Access: (0644/-rw-r--r--) Uid: ( 1000/ oliver)  Gid: ( 1000/ oliver)  
Access: 2011-03-08 18:11:48.000000000 +0100  
Modify: 2011-03-06 19:28:22.000000000 +0100  
Change: 2011-03-06 21:31:15.000000000 +0100
```

- Man beachte die alternative Schreibweise 0644 !

Berechtigungen kennen und verstehen (4)

Dateiberechtigungen numerisch anzugeben ist nicht notwendig, aber einfacher und erleichtert die Modifikation.

	Vollzugriff	Lesen/Schreiben	Lesen/Ausführen
Textnotation	r w x	r w -	r - x
Wertigkeit	4 2 1	4 2 1	4 2 1
Numerisch	7	6	5

	Nur Lesen	Nur Schreiben	Nur Ausführen
Textnotation	r - -	- w -	- - x
Wertigkeit	4 2 1	4 2 1	4 2 1
Numerisch	4	2	1

Berechtigungen verändern

- **chmod** (change mode) Ändert Zugriffe auf Dateien.

`chmod [ugo][+-][rwx] <datei/verz>`

Bsp.: `chmod go-rwx <datei/verz>`

Gruppe/andere dürfen nicht lesen/schreiben/ausführen

`chmod o+x <datei/verz>`

Andere dürfen Dateien ausführen oder Verzeichnisse betreten.

- **chown** (change owner) Ändert den Besitzer oder auch die Gruppenzugehörigkeit.

`chown <Besitzer> <Datei/Verz>` Besitzer

`chown :<Gruppe> <Datei/Verz>` Gruppe


`chown <Besitzer>:<Gruppe> <Datei/Verz>` beides

Den Besitzer darf nur Root ändern !!!

- **chgrp** (change group) Ändert die Gruppenzugehörigkeit der Datei.

`chgrp <gruppe> <datei/verz>`





Interessante Befehle und Tastaturkürzel, die das tägliche Leben erleichtern.

- Die Tab-Taste  zum Vervollständigen von Befehlen, Dateinamen und Verzeichnissen.
- Pipelining mit |

Bsp.: `ls | less`

Hier wird die Ausgabe von `ls` an den Befehl `less` weitergegeben, der die Ausgabe von `ls` seitenweise ausgibt.

Alte Befehle wieder finden

- Mit den Pfeiltasten  und 
- **history** listet die letzten 500 benutzten Befehle auf.
- **!n** führt den n-ten Befehl aus.
- **!!** führt den letzten Befehl aus.
- **!-n** führt n-letzten Befehl aus.
- + <string> sucht nach dem letzten Befehl, der mit <string> beginnt.

System aktuell halten (1)

Um das Betriebssystem aktuell zu halten bzw. neue Software zu installieren, kann folgender Befehl verwendet werden:

- `apt-get <Option>`

Folgende Optionen sind möglich:

- `update` Aktualisiert die Paketindexdatei.
- `upgrade` Installiert die aktuelle Version der installierten Pakete.
- `dist-upgrade` Berücksichtigt zusätzlich Abhängigkeiten zu neueren Paketen.
- `install <Paket(e)>` Installiert die angegebenen Pakete.
- `remove <Paket(e)>` Löscht die angegebenen Pakete.
(Konfigurationsdateien bleiben erhalten)
- `purge <Paket(e)>` Löscht alles von den Paketen.
- `clean` Löscht die heruntergeladenen Pakete.
- `autoclean` Löscht die Pakete, die nicht mehr heruntergeladen werden können.
- `autoremove` Löscht Pakete, die heruntergeladen wurden, um Abhängigkeiten zu erfüllen, und dann nicht mehr benötigt werden.

Um Pakete zu suchen, hilft dieser Befehl weiter:

- **apt-cache <Option>**

Folgende Optionen sind möglich:

- **showpkg <Paket(e)>** Zeigt Informationen zu den angegebenen Paketen an.
- **show <Paket(e)>** Zeigt die Paketdatensätze für die Pakete an.
- **search <Ausdruck>** Sucht Paketnamen und Beschreibungen in denen <Ausdruck> auftritt und gibt die Paketnamen aus.

System aktuell halten (3)

Eine alternative zu apt-get und apt-cache ist:

- `aptitude <Option> <Paket(e)>`

Mögliche Optionen sind:

- `update`
- `upgrade`
- `install <Paket(e)>`
- `remove <Paket(e)>`
- `purge <Paket(e)>`
- `search <Paket(e)>`
- `show <Paket(e)>`
- `clean`
- `autoclean`

Wer sucht, der findet. (1)

Dateien suchen mit:

- **find**

find <Startverzeichnis> <Auswahlkriterien> <Aktion>

Der Befehl sucht rekursiv im Verzeichnisbaum nach einer Datei.

Wichtige Auswahlkriterien:

- **-name** Sucht nach passenden Dateinamen.
- **-user** Sucht nach Dateien, die dem angegebenen Benutzer gehören.
- **-group** Sucht nach Dateien, die der angegebenen Gruppe gehören.
- **-type** Sucht nach dem angegebenen Datentyp
- **-o** (or) Eine der Bedingungen muss zutreffen.
- **-a** (and) Alle Bedingungen müssen zutreffen.

Bsp.: `find . \(-type d -o -name 'A*' \) -print`
`find /home -user Test -exec ls -l '{}'` \;

Wer sucht, der findet. (2)

- **locate** bzw. **slocate**

Bevor der Befehl `locate` Wirkungen zeigt, muss die dazugehörige Datenbank mit `updatedb` (als Root) erstellt werden.

Bsp.: `sudo updatedb`
`locate Konsole.png`

- **which**

Der Befehl `which` sucht nach ausführbaren Dateien (also Programmen)

- **whereis**

Der Befehl `whereis` sucht nach Programmen und nach manpages.

- **grep**

`grep <Suchindex> <Datei>`

Bsp.: `find Frosch frosch.txt`

Findet alle Zeilen, die das Wort Frosch enthalten (Froschkönig).

`find '\< Frosch\>' frosch.txt`

Findet alle Zeilen, die genau das Wort „Frosch“ enthalten.

Es gibt unterschiedliche Konsolen-Editoren unter Linux.

- vim Vi-Improved
 - Arbeitet mit unterschiedlichen Modi
 - Kommandomodus
 - Einfügen und Editieren
 - visual-Mode
- emacs Editor Macros
 - Macros für verschiedene Anwendungen
 - Email-Client
 - Browser
 - MP3-Player
 - Spiele
- nano Nano's another editor
- ... und für jeden Geschmack eine Menge weiterer

Vielen Dank für die Aufmerksamkeit.

