

Linux Basiswissen

Agenda

1. Verzeichnisse und Dateitypen
 - [Verzeichnisaufbau](#)
 - [Dateitypen](#)
2. Basisbefehle
 - [In den Root-Benutzer wechseln](#)
 - [Wo bin ich ?](#)
 - [Praktische Ausgabe von langen Seiten - less](#)
 - [Datei anlegen - touch](#)
 - [Autovervollständigen * und tab](#)
 - [Welches Programm wird verwendet](#)
3. Dateien und Verzeichnisse
 - [Mit cd im System navigieren](#)
 - [Verzeichnisse in Listenansicht mit versteckten Dateien anzeigen](#)
 - [Inhalt in Datei schreiben und anhängen](#)
 - [Verzeichnisse und Dateien löschen](#)
 - [Kopieren/Verschieben/Umbenennen von Dateien und Files](#)
4. Prozesse
 - [Prozesse anzeigen - ps/pstree -p](#)
5. Benutzer, Gruppen und Rechte
 - [Rechte](#)
 - [Dateien für Benutzer und Gruppen](#)
 - [Benutzer anlegen](#)
 - [sudo Benutzer erstellen](#)
6. Dateimanipulation/Unix Tools
 - [cat/head/tail-Beginn/Ende einer Datei anzeigen](#)
 - [zcat - Inhalte einer mit gzip komprimierten Datei anzeigen](#)
 - [wc - Zeilen zählen](#)
 - [Bestimmte Zeilen aus Datei anzeigen - grep](#)
7. Logs/Loganalyse
 - [Logfile beobachten](#)
8. Variablen
 - [Setzen und verwenden von Variablen](#)
9. Dienste/Runlevel(Targets verwalten)
 - [Die wichtigsten systemctl/service](#)
10. Partitionierung und Filesystem
 - [parted and mkfs.ext4](#)
11. Boot-Prozess und Kernel
 - [Grub konfigurieren](#)
 - [Kernel-Version anzeigen](#)
 - [Kernel-Module laden/entladen/zeigen](#)
12. Hilfe
 - [Hilfe zu Befehlen](#)
13. Literatur
 - [Literatur](#)

Verzeichnisse und Dateitypen

Verzeichnisaufbau

/etc

- Verzeichnis für Konfigurationsdateien

/dev

- Devices (Alle Gerätedateien - Ein- und Ausgabegeräte, wie bspw. Festplatten, Mouse)

/mnt

- früher viel verwendet:
- für händisches Einhängen gedacht (per Hand mounten)

/media

- das neue / moderne (wird heutzutage meistens verwendet)
- Verzeichnis für automatisch eingehängte Devices (z.B. usb-stick)

/opt

- Große Softwarepaket (z.B. LibreOffice, OpenOffice, Dritt-Anbieter)

/boot

- Files for booting (e.g. kernel, grub.cfg, initial ramdisk)

/proc

- Schnittstelle zwischen Kernel und User-Space (für Programme, Benutzer)
- Kommunikation erfolgt über Dateien

/root

- Heimatverzeichnis des root-Benutzers

/run

- Dateien mit Prozess-ID für laufenden Services
- um diese gut beenden zu können

/tmp

- Temporäre Dateien
- Löschen von Dateien kann unter /etc/tmpfiles.d verwaltet werden (erfolgt von systemd auf Tagesbasis)

/sys

- wie proc
- Schnittstelle zwischen Kernel und User-space

/var (=variable daten)

- Hier liegen Daten, die sich häufig ändern
- Log-Dateien, Datenbanken, Spool-Dateien, Cache-Dateien

/lib

- Bibliotheken (.so, .ko) wie unter Windows *.dll's

/sbin

- Programme zur Systemadministration

/bin

- Normale Programme für alle (executables)

Dateitypen

Wo ?

- Erste Spalte bei ls -la

Welche ?

```
- file  
d directory  
l symbolischer Link  
c Character-Device (Eingabegerät: Zeichenorientiert z.B. Tastatur)  
b Block-Device (Ausgabegerät): Blockorientiert, z.B. Festplatte)
```

Basisbefehle

In den Root-Benutzer wechseln

```
## einloggen als normaler Benutzer z.B. benutzer: kurs  
sudo su -  
## eingeben des Passworts des benutzer
```

Wo bin ich ?

```
## 1. Ich erkenne es am prompt (Beginn der Zeile )

## pwd - Print working directory
pwd
```

Praktische Ausgabe von langen Seiten - less

Open a file with less

```
##  
less /etc/services  
  
## Why ?  
## Leichtere Navigation
```

Pipen mit less (ausgabe an less schicken)

```
ls -la | less  
cat /etc/services | less
```

Suchen in less

```
##Innerhalb von less  
/suchbegriff + RETURN  
## nächstes Suchergebnis  
n
```

Springen ans Ende/an den Anfang

```
## Innerhalb von less  
## ans Ende  
G  
## an den Anfang  
lg  
## zu einer bestimmten Zeile (Zeile 5)  
5g
```

In die Hilfe rein

```
h  
## wieder raus  
q
```


Datei anlegen - touch

```
touch dateiname
```

Autovervollständigen * und tab

Autovervollständigen *

```
## show all entries in directory starting with tod
## * = zero or more characters
echo tod*
## tod todo todotext
```

Autovervollständigen tab

```
echo tod <TAB><TAB> # bei mehreren Einträgen
echo todol<TAB> # bei einem weiteren Eintrag
```

Welches Programm wird verwendet

```
## Sucht in der Pfad-Variablen $PATH nach dem programm  
## und zeigt ersten Fund --> d.h. dieses Programm würde ausgeführt  
which false
```

Dateien und Verzeichnisse

Mit cd im System navigieren

Ins Heimatverzeichnis und Wurzelverzeichnis (C: unter Windows) wechseln

```
## Ins Heimatverzeichnis wechseln
## cd ohne alles
cd

## Ins Wurzelverzeichnis
cd /
```

Wie in ein Verzeichnis wechseln (relativ und absolut)

```
## relativ - nur in ein Unterverzeichnis meines bestehenden Verzeichnisses
cd etc

## absolut - wechselt dort rein, egal wo ich bin
cd /etc
```

Verzeichnisse in Listenansicht mit versteckten Dateien anzeigen

```
ls -la
```

Inhalt in Datei schreiben und anhängen

Inhalte in Datei schreiben / anhängen

```
cd /home/kurs
## eingefügt am anfang, überschreibt alte Inhalte
ls -la > todo
## angehängt
echo "hans hat durst" >> todo
```

Verzeichnisse und Dateien löschen

Dateien und Verzeichnisse löschen

```
## bei symbolischen Links wird nur der symbolische Link und nicht die Datei gelöscht  
rm symlink  
## Datei löschen  
rm dateiname  
## Verzeichnis löschen  
rm -r verzeichnis
```

Kopieren/Verschieben/Umbenennen von Dateien und Files

Dateien umbenennen, verschieben, kopieren

```
## wenn Zielverzeichnis nicht existiert -> Fehler !
cp -a todo.txt /dokumente/
## wenn zielverzeichnis nicht existiert, wird dokumente2 erstellt als file - > Achtung
!!
cp -a todo.txt /dokumente2

## umbenennen
mv datei1 neuernamedatei1

## verschieben
mv datei1 /dokumente
```

Rechte behalten bei kopieren

```
## -a macht das
cp -a todo.txt todoneu.txt

## ohne -a werden symbolische links aufgelöst und die Rechte des ausführenden gesetzt
cp ab cd
```


Prozesse

Prozesse anzeigen - ps/pstree -p

Prozesse anzeigen

```
ps -ef  
ps aux # x alle Prozesse anzeigen, die nicht an ein Terminal gebunden sind
```

systemctl (läuft Dienst)

```
systemctl status sshd
```

Prozeßbaum anzeigen (meist nicht für die Praxis notwendig)

```
pstree -p
```

Benutzer, Gruppen und Rechte

Rechte

Aufbau triple

```
kurs@ubuntu2004-101:~$ # rwx | rw- | r--  
kurs@ubuntu2004-101:~$ # u   g   o  
kurs@ubuntu2004-101:~$ # 421 | 42- | 4--  
kurs@ubuntu2004-101:~$ #  7  |  6  |  4
```

Berechtigungen mit Symbolen setzen

```
chmod g+w,o+r testfile
```

Dateien für Benutzer und Gruppen

```
cd /etc  
cat passwd  
cat shadow  
cat group
```

Benutzer anlegen

Benutzer anlegen (auf Ubuntu)

```
## for shell script
useradd

## for admins interactive
adduser
```

sudo Benutzer erstellen

Benutzer zum Sudo benutzer machen

```
adduser newuser
usermod -aG sudo newuser
### testing
su - newuser
groups # see if we are in groups sudo
id # shows the same but more info
## need to enter password here
sudo su -
```

Dateimanipulation/Unix Tools

cat/head/tail-Beginn/Ende einer Datei anzeigen

cat mit Zeilennummer

```
cat -n /etc/services
```

Die ersten -x Zeilen anzeigen

```
## ersten 10 Zeilen anzeigen  
head /etc/services  
  
## Ersten 20 Zeilen  
head -n 20 /etc/services
```

Die letzten -x Zeilen anzeigen

```
## die letzten 10 Zeilen  
tail /etc/services  
  
## die letzten 40 Zeilen  
tail -n 40 /etc/services
```

Ausgabe der letzten Zeilen und ausgabe in Datei

```
cd /var/log  
tail -n 100 syslog.1 >> fehlerlog  
cat fehlerlog
```

zcat - Inhalte einer mit gzip komprimierten Datei anzeigen

wc - Zeilen zählen

Datei

```
wc -l /etc/services
```

Zeilen aus Befehl

```
ls -la | wc -l
```

Bestimmte Zeilen aus Datei anzeigen - grep

Beispiele

```
## alle Zeilen in den tcp vorkommt
cat /etc/services | grep tcp
## alle Zeilen in denen tcp nicht vorkommt
cat /etc/services | grep -v tcp
## alle Zeilen in denen tcp nicht vorkommt
## egal ob gross oder klein geschrieben.
cat /etc/services | grep -iv TCP

cat /etc/services | grep '#'
cat /etc/services | grep "#"
cat /etc/services | grep "^#"
## alle Zeilen, die am Anfang der Zeile kein # haben
cat /etc/services | grep -v "^#"
cat /etc/services | grep -v "^#" > /root/services
cat /etc/services | grep -v "^#" | head -n 20

cat /etc/services | grep -v "s$"
## alle Zeilen die als letztes Zeichen ein s haben
cat /etc/services | grep "s$"
```

Recursive Suchen (grep -r)

```
grep -r "PermitRootLogin" /etc
```


Logs/Loganalyse

Logfile beobachten

```
## Terminal 1
tail -f /var/log/syslog

## Terminal 2 - write to logfile e.g.
logger meine_nachricht
```

Variablen

Setzen und verwenden von Variablen

```
DATEINAME=/etc/services
echo $DATEINAME

# Werte hochzählen
ZAHL=4
let ZAHL=ZAHL+1
echo $ZAHL

cat $DATEINAME
# wird nicht der Inhalt verwendet sondern der Name $DATEINAME
cat '$DATEINAME'
cat "$DATEINAME"

# Befehl ausführen und Rückgabewert anzeigen
date
echo $?

# Wert aus ausgeführtem Befehl in Variable schreiben
DATUM=$(date)
echo $DATUM
echo $DATUM >> /var/log/datumslog
```

Dienste/Runlevel(Targets verwalten)

Die wichtigsten systemctl/service

systemctl Beispiele

```
## Status eines Dienstes überprüfen
service sshd status
systemctl status sshd

## Wie heisst der Dienst / welche Dienste gibt es ?
systemctl list-units -t service
## für apache
systemctl list-units -t service | grep ^apache
## die Abkürzung
systemctl -t service | grep ^apache

## Dienst aktivieren
systemctl enable apache2
## Ist Dienst aktiviert
systemctl is-enabled apache2
enabled
echo $?
0 # Wenn der Dienst aktiviert ist

## Dienst deaktivieren (nach Booten nicht starten)
systemctl disable apache2
systemctl is-enabled
disabled
echo $?
1 # 1 wenn nicht aktiviert

## Rebooten des Servers
## verweist auf systemctl
reboot
systemctl reboot
shutdown -r now

## Halt (ohne Strom ausschalten)
halt
systemctl halt
shutdown -h now

## Poweroff
poweroff
systemctl poweroff
```

Welche Dienste sind aktiviert/deaktiviert

```
systemctl list-unit-files -t service
```

Dienstekonfiguration anzeigen

```
systemctl cat sshd.service
```

Dienste bearbeiten

```
systemctl edit sshd.service
## Dann eintragen
[Unit]
Description=Jochen's ssh-server
## Dann speichern und schliessen (Editor)

systemctl daemon-reload
systemctl status
```

Targets (wechseln und default)

```
## Default runlevel/target auslesen
systemctl get-default
## in target wechseln
systemctl isolate multi-user
## Default target setzen (nach start/reboot)
systemctl set-default multi-user
```

Alle Target anzeigen in die ich reinwechseln kann (isolate)

```
## Ubuntu
grep -r "AllowIsolate" /lib/systemd/system
/lib/systemd/system/reboot.target
...
...
...
systemctl isolate reboot.target
```

Dienste maskieren, so dass sie nicht gestartet werden können

```
systemctl mask apache2
## kann jetzt gestartet werden
systemctl start apache2

## de-maskieren
systemctl unmask apache2
## kann wieder gestartet werden
systemctl start apache2
```

systemctl Cheatsheet

- https://access.redhat.com/sites/default/files/attachments/12052018_systemd_6.pdf

Partitionierung und Filesystem

parted and mkfs.ext4

Walkthrough

```
## Schritt 1: Platte in virtualbox oder gui-interface anlegen

## Schritt 2: Platte identifizieren
lsblk

## Schritt 3: Platte partitionieren
mkpart /dev/sdb1
mklabel gpt
mkpart data2 ext4 2048s 500M # data2 ist name der Partition bei gpt
quit

## Schritt 4: Partition formatiert
lsblk # Partition identifiziert
mkfs.ext4 /dev/sdb1

## Schritt 5: Mount-Punkt erstellen
mkdir /mnt/platte

## Schritt 6: einhängen und aushängen
mount /dev/sdb1 /mnt/platte
umount /mnt/platte

## Schritt 7: Persistent konfigurieren
## Eintragen in /etc/fstab
/dev/sdb1 /mnt/platte ext4 defaults 0 0

## Schritt 8: Test, ob fstab gut ist (keine Fehler)
mount -av # v steht für geschwätzig.

## Wenn das klappt: Schritt 9
reboot
```

Boot-Prozess und Kernel

Grub konfigurieren

Walkthrough

```
## Step 1
## z.B. timeout hochsetzen, wie lange er mit Booten im Bootmenu wartet
cd /etc/default
vi grub
### make wanted changes
##GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=5

## Step 2
update-grub

## Step 3 - reboot

## When grub menu appears enter arrow-down arrow-up ONCE
## Dann zählt er nicht weiter runter und bootmenu bleibt stehen.

## Mit e kann man einen boot-eintrag für den nächsten Boot ändern

## Ändern und dann CTRL bzw. STRG + x für das Booten nach Änderung

## Step 4 - be happy
```

Kernel-Version anzeigen

```
uname -a
```

Kernel-Module laden/entladen/zeigen

Walkthrough

```
## show kernel modules
lsmod
## kernel - module entladen
modprobe -r psmouse
lsmod | grep psmouse # now not present
## damit wieder laden
modprobe psmouse
lsmod | grep psmouse # now present
```

Wo leben die Kernel - Module

```
### kernel version is used, find out kernel version with
uname -a

cd /lib/modules/5.4.0-66-generic

## e.g. psmouse
find /lib/modules -name psmouse*
/lib/modules/5.4.0-66-generic/kernel/drivers/input/mouse/psmouse.ko
```


Hilfe

Hilfe zu Befehlen

Möglichkeiten der Hilfe

```
## anhand von ps
```

```
vi -h
```

```
ps --help
```

```
man ps
```

```
info ps
```

Literatur

Literatur

Literatur

- [Linux Grundlagen für Anwender und Administratoren](#)
- [Linux Systemadministration I für Anwender und Administratoren](#)
- [Alle Unterlagen](#)

Cheatsheet

- [Cheatsheet bash](#)