

Linux Basiswissen

Agenda

1. Distributionen
 - [Überblick](#)
2. Verzeichnisse und Dateitypen
 - [Verzeichnisaufbau](#)
 - [Dateitypen](#)
3. Administration / Tipps & Tricks
 - [IP-Adresse herausfinden](#)
4. Basisbefehle
 - [In den Root-Benutzer wechseln - sudo](#)
 - [Wo bin ich ?](#)
 - [Praktische Ausgabe von langen Seiten - less](#)
 - [Datei anlegen - touch](#)
 - [Autovervollständigen * und tab-Taste](#)
 - [Welches Programm wird verwendet](#)
5. Erweiterte Befehle (Nice to have)
 - [Alias Befehle anzeigen](#)
 - [Dateien und Ordner vergleichen - diff](#)
6. Dateien und Verzeichnisse
 - [Mit cd im System navigieren](#)
 - [Verzeichnisse in Listenansicht mit versteckten Dateien anzeigen -> ls -la](#)
 - [Inhalt in Datei schreiben und anhängen](#)
 - [Verzeichnisse anlegen](#)
 - [Verzeichnisse und Dateien löschen](#)
 - [Kopieren/Verschieben/Umbenennen von Dateien und Files](#)
 - [Arbeiten mit vi](#)
7. Prozesse
 - [Prozesse anzeigen - ps/pstree -p](#)
8. Benutzer, Gruppen und Rechte
 - [Rechte](#)
 - [Dateien für Benutzer und Gruppen](#)
 - [Benutzer anlegen](#)
 - [Benutzer löschen \(debian\)](#)
 - [sudo Benutzer erstellen](#)
9. Dateimanipulation/Unix Tools
 - [Anfang oder Ende einer Datei/Ausgabe anzeigen](#)
 - [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](#)
 - [Erweiterte Suche mit Grep](#)
 - [Finden von files nach Kriterien - find](#)
10. Logs/Loganalyse
 - [Logfile beobachten](#)
 - [Dienste debuggen](#)
 - [Rsyslog](#)
 - [Journal analysieren](#)
11. Bash: Variablen, Kommandosubstitution und Quotes

- [Kommandos ausgeben und weiter verwenden](#)
 - [Setzen und verwenden von Variablen](#)
 - [Quoting](#)
 - [Die history](#)
12. Dienste/Runlevel(Targets verwalten)
- [Systemd Überblick](#)
 - [Die wichtigsten systemctl/service](#)
 - [Script mit systemd verwalten und EnvironmentVariablen](#)
 - [Systemd Service endless loop](#)
 - [Systemctl - timers](#)
 - [systemctl - timers - Übung](#)
 - [Gegenüberstellung service etc/init.d/ systemctl](#)
13. Administration
- [Dienste debuggen](#)
 - [Neue Partition mit lvm](#)
 - [Verzeichnisse backup/restore mit tar](#)
14. Apache
- [SSL - LetsEncrypt mit Ubuntu 22.04](#)
15. Partitionierung und Filesystem
- [parted and mkfs.ext4](#)
16. Boot-Prozess und Kernel
- [Hintergründe zum Starten des Systems](#)
 - [Grub konfigurieren](#)
 - [Kernel Params beim Booten übergeben](#)
 - [Kernel-Version anzeigen](#)
 - [Kernel-Module laden/entladen/zeigen](#)
17. Hilfe
- [Hilfe zu Befehlen](#)
18. Grafische Oberfläche und Installation
- [Gnome unter Ubuntu installieren](#)
 - [X-Server - Ausgabe auf Windows umleiten](#)
 - [Installations-Images-Server](#)
19. Wartung, Sicherung und Aktualisierung
- [Aktualisierung des Systems](#)
 - [Paketmanager apt/dpkg](#)
 - [Archive runterladen und entpacken](#)
 - [Mehrere Versionen eines Programms z.B. php \(cli\) verwalten](#)
 - [Verzeichnisse in archiven sichern - tar](#)
20. Absicherung System
- [ssh absichern](#)
21. Firewall und ports
- [ufw \(uncomplicated firewall\)](#)
 - [firewalld](#)
 - [Scannen und Überprüfen mit telnet/nmap](#)
22. Netzwerk/Dienste
- [Wie Netzwerk einrichten in unterschiedlichen Distros](#)
 - [Hostname setzen](#)
 - [netplan unter Ubuntu](#)
 - [Netzwerk unter SLES - OpenSuSE - wicked](#)
 - [Auf welchen Ports lauscht mein Server - Isot](#)
23. Bash/Bash-Scripting

- [Einfaches Script zur Datumsausgabe](#)
- [Script Beispiel](#)
- [Ausführen/Verketteten von mehreren Befehlen](#)
- [Vordefinierte Variablen z.B \\$0](#)
- [Funktionen in der bash](#)
- [Best practice structure bash - scripts](#)
- [Neue Umgebungsvariable setzen](#)

24. Timers/cronjobs

- [Cronjob - hourly einrichten](#)
- [cronjob \(zentral\) - crond](#)

25. Übungen

- [Übung Verzeichnis](#)
- [Übung Dienste](#)
- [Übung Umleitung mit Variable](#)
- [Übung user/password](#)

26. Literatur

- [Literatur](#)
- [Cheatsheet Commandline](#)
- [Wo finde ich Hilfe im Internet](#)
- [RDP-Client unter Windows](#)
- [Linux Malware Scanner](#)

Backlog

1. Erweiterte Befehle (Nice to have)

- [Welche Bibliotheken verwendet ein ausführbares Programm](#)

2. Tools/Verschiedens

- [Remote Desktop für Linux / durch Teilnehmer getestet](#)
- [Warum umask 002 und 0002 ? - Geschichte](#)
- [lokale Mails installieren](#)
- [Debian/Ubuntu - deb - Paket entpacken](#)
- [Geänderte Dateien zu anderem Server schicken](#)

3. Prozesse

- [Prioritäten und NiceNess](#)

4. Netzwerk

- [IP-Adresse von DHCP-Server holen \(quick-and-dirty\)](#)

5. Digitalocean

- [Script zum Aufsetzen eines Server mit Docker](#)

Distributionen

Überblick

Multi-Purpose - Distributionen (Ideal zum Starten)

Redhat-Familie

```
Centos
Redhat.  - rpm / (yum / dnf) (Support-Vertrag)
Fedora
Rocky Linux
Scientific Linux
```

Debian Familie

```
Debian (Stiftung)
Ubuntu. - dpkg / apt (Support-Vertrag möglich)
Mint
```

SuSE - Familie

```
SLES (SuSE Linux Enterprise)
OpenSuSE
```

Was hat die jeweilige Familie gemeinsam ?

```
Installer
Grafische Oberfläche (GNOME)
Bestimmte Teilmenge von Anwendungen / Features
Paket-Management-System

Installer, Grafische Oberfläche, Paket-Management, Anwendungen und Kernel -> nennt man ->
Distribution
```

Varianten Ubuntu

```
Ubuntu Server
Ubuntu Desktop

-> immer die Ubuntu Server -> Netzwerk serverorientiert
```

Varianten Debian

```
Immer: Debian Minimal Version verwenden
(Wir fangen mit dem kleinsten gemeinsamen Nenner an:)
https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/debian-11.3.0-amd64-netinst.iso
```

Redhat

```
RHEL 8 (Redhat Enterprise Linux)
Rocky Linux 8 (Stand 2022)
```

SuSE

```
Subscription: SLES 15 (SuSE Linux Enterprise Server)
```

Distris zur Sicherheitsüberprüfung / Hacken

```
Kali Linux  
Parrot. - Distributionen zum Hacken
```

Live-DVD (Linux ohne Installation)

- Knoppix - Live DVD - brauche nicht installieren

Spezial-Linuxe, z.B. für Router

```
OpenWRT  
DDWRT
```

Seite mit Übersicht aller Linux-Distros

- <https://distrowatch.com/>

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, initital 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
```

Administration / Tipps & Tricks

IP-Adresse herausfinden

```
ip a
## noch schöner - ausgabe ist schlanker
ip -br a
```

Basisbefehle

In den Root-Benutzer wechseln - sudo

```
## einloggen als normaler Benutzer z.B. benutzer: kurs (wenn ich unter kurs eingeloggt bin)
sudo su -
## eingeben des Passworts des Benutzers
```

```
## oder gleichbedeutend:  
sudo -i
```

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  
## voriges Suchergebnis  
N
```

Springen ans Ende/an den Anfang

```
## Innerhalb von less  
## ans Ende  
G  
## an den Anfang  
1g  
## zu einer bestimmten Zeile (Zeile 5)  
5g
```

In die Hilfe rein

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

Datei anlegen - touch

```
touch dateiname
```

Autovervollständigen * und tab-Taste

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

which verwenden

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

Pfad ausgeben

```
env
echo $PATH
```

Erweiterte Befehle (Nice to have)

Alias Befehle anzeigen

Alias anzeigen

```
## keine wirkliche Befehle, sondern nur andere Schreibweise/Abkürzungen
## kann u.U. so auf anderen Distros nicht vorhanden sein
alias
```

Alias anlegen

```
## in der Session
alias hallo='ls -la'

## persistent bei jedem aufruf einer bash
## unter ubuntu 20.04.
## bash_aliases wird in ~/.bashrc geladen
echo "alias hallo2='ls -la'" > ~/.bash_aliases

## zum Testen
## 1. Entweder
## Neue bash starten
bash
## Sourcen zum einmaligen Testen
```



```
source ~/.bash_aliases
```

Dateien und Ordner vergleichen - diff

Dateien vergleichen

```
diff dateia /path/zu/dateib
```

Ordner vergleichen

```
diff -r /etc /etc2
```

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 des Filesystems wechseln // Windows -> C:\
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

```
ls -la
```

Inhalt in Datei schreiben und anhängen

Inhalte in Datei schreiben / anhängen

```
cd /home/kurs
## Alternative 1
## cd # wechselt auch ins Heimatverzeichnis
## Alternative 2
## cd ~

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

Verzeichnisse anlegen

Einzelne Verzeichnisse anlegen

```
## Verzeichnis Dokumente anlegen im aktuellen Verzeichnis
cd
mkdir dokumente

## absolut verzeichnis anlegen
## Wird dann im Wurzelverzeichnis angelegt als root
## als kurs-benutzer hätte ich dort keine Berechtigung
sudo mkdir /docs
```

Verzeichnisstruktur anlegen

```
cd
## Elternverzeichnisse werden automatisch angelegt
mkdir -p dokumente/projekt/plan
```

Verzeichnisstruktur anzeigen

```
sudo apt install tree
tree dokumente

## or /etc
tree /etc | less
```

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
```

Mehrere Dateien löschen

```
cd
touch datei1 datei2 datei3
echo datei*
rm datei*
```

Symbolische Links löschen (Verhalten)

```
cd
touch woche.txt
ln -s woche.txt woche1.txt
## file woche.txt is still present
rm woche1.txt
ls -la
```

```
## Symbolischen Link erneut setzen
ln -s woche.txt woche1.txt
## Symbolischer Link danach kaputt
rm woche.txt
ls -la
## woche1.txt nicht aufrufbar, da der symbolische Link ins Leere zeigt.
cat woche1.txt
```

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 in Verzeichnis
mv datei1 /dokumente/
## besser als:
## mv datei1 /dokumente
## weil hier die Datei dokumente angelegt wird, wenn der Ordner /dokumente nicht existiert !!
```

Rechte behalten bei kopieren

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

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

## Verzeichnisse kopieren
cp -a /etc /etc3
```

Arbeiten mit vi

vim installieren (falls nicht installiert)

```
## SLES
zypper install vim
```

```
## Ubuntu
apt install vim
```

Zeilennummern aktivieren für meinen User

```
cd
vi .vimrc
## eitragen
set number
```

Wichtigste Aktionen

```
1. # Öffnen eine neuer Datei mit vi
vi dateiname

2. # Schreiben in der Datei
i # <- i-Taste drücken

3. # Es erscheint unten in der Zeile
# -- INSERT --

4. # Nun können Sie etwas hineinschreiben

5a. Beenden ohne Speichern (wenn geänderter Inhalt vorhanden ist
ESC + :q! # ESC Taste drücken, dann : und q! und enter

5b. Oder: Speichern und schliessen
ESC + :x # ESC Taste drücken, dann : und w und enter
```

Virtual Mode

```
v Zeichenweise markieren einschalten
V Zeilenweise markieren einschalten
STRG + v Blockweise markieren

## mit Cursortasten auswählen / markieren
## Dann:
x # Löschen des markierten Bereichs
```

Zeilen löschen im Normalmodus (Interactiver Modus)

```
ESC + dd # eine Zeile löschen
## letzte Aktion rückgängig machen
ESC + u # eigentlich reicht 1x Escape
## mehrere Zeilen löschen z.B. 1000
ESC + 1000dd # ESC - Taste drücken, dann 1000 eingeben, dann dd (sie sehen die 1000 nicht auf dem Bildschirm)
```

Neues Fenster und Fenster wechseln

```
## innerhalb von vi
ESC + : -> vsplit # aktuelles Fenster wird kopiert
## Fenster wechseln
ESC + : wincmd w
## oder
STRG + w w
```

Cheatsheet

http://www.atmos.albany.edu/daes/atmclasses/atm350/vi_cheat_sheet.pdf

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

Arten

```
r = Lesen
w = Schreiben
x = Ausführen
```

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

## rwx | rw- | r--
## u g o
## 421 | 42- | 4--
## 7 | 6 | 4
```

Bedeutung Oktalzahlen

```
## rwx
## r = 4, w = 2, x = 1
```

Beispiele

```
- w -
0 2 0 = 020
```

Berechtigungen mit Symbolen setzen

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

Dateien für Benutzer und Gruppen

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

kurs@ubuntu2004-104:/etc$ ls -la passwd shadow group
-rw-r--r-- 1 root root 1097 Mar 10 10:06 group
-rw-r--r-- 1 root root 3164 Mar 10 10:06 passwd
-rw-r----- 1 root shadow 1838 Mar 10 10:06 shadow
```

Benutzer anlegen

Benutzer anlegen (auf Ubuntu)

```
## for shell script
useradd

## for admins interactive
adduser
```

Benutzer löschen (debian)

```
## Remove training user and his file in home-directory
deluser --remove-home training
```

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

Anfang oder Ende einer Datei/Ausgabe anzeigen

Die ersten 10

```
## die ersten 10 Zeilen einer Datei anzeigen
head /etc/services
## Alternative 1
cat /etc/services | head

## die letzten 10 Zeilen
tail /etc/services
cat /etc/services | tail
```

```
## einer ausgabe // erste 10 Zeilen eines Verzeichnislistings
ls -la | head
```

Die ersten 20

```
head -n 20 /etc/services
head -n20 /etc/services
head -20 /etc/services
head --lines=20 /etc/services
```

Die letzten 20

```
tail -n 20 /etc/services
tail -n20 /etc/services
tail -20 /etc/services
tail --lines=20 /etc/services
```

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

```
zcat services.gz
```

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

Ergebnis und 1 Zeile danach

```
apt search apache | grep -A 1 ^apache
## Alternativ für -B 10 davor (10 Zeilen davor)
```

Anzahl der Vorkommen anzeigen

```
ps aux | grep -c apache
```

Recursive Suchen (grep -r) - Schweizer Taschenmesser

```
grep -r "PermitRootLogin" /etc

## case insensitive # egal ob gross oder klein
grep -ir "LISTEN" /etc

## Mit Zeilennummer
grep -nr "PermitRootLogin" /etc
```

Erweiterte Suche mit Grep

Nach einzelnen Wort suchen (Wort muss so vorkommen)


```
cat /etc/services | grep -i -w 'protocol'
```

Eines der Begriffe soll vorkommen

```
## Achtung, unbedingt -E für extended regex verwendet
cat /etc/services | grep -E 'protocol|mysql'
```

Eines der Wort soll am Anfang der Zeile vorkommen

```
## egrep ist das gleiche wie grep -E
egrep -i '^(mysql|Maira)' /etc/services
```

x-Zeilen vor bzw. nach "Finde-(Grep-)" - Ergebnis anzeigen

```
## -A x-Zeilen danach, z.B. -A 4 --> 4 Zeilen danach
## -B x-Zeilen davor
egrep -A 4 -B 4 -i '^(mysql|Maira)' /etc/services
```

Einzelne Zeichen als Suchmuster nehmen

```
## 0, dann zwei beliebige Zeichen, dann tcp
grep '0..tcp' /etc/services
## 0, dann ein beliebiges Zeichen, dann tcp
grep '0.tcp' /etc/services
```

Tatsächlich eine Punkt suchen

```
## /root/dateinamen
hans.txt
hans1txt
peter.txt

grep 'hans\.txt' /root/dateinamen

root@ubuntu2004-101:/etc# grep 'hans\.txt' /root/dateinamen
hans.txt
root@ubuntu2004-101:/etc# grep 'hans.txt' /root/dateinamen
hans.txt
hans1txt
```

Einzelne Zeichen sollen vorkommen

```
root@ubuntu2004-101:~# echo "Klaus" >> /root/namen
root@ubuntu2004-101:~# echo "klaus" >> /root/namen
root@ubuntu2004-101:~# grep '[kK]l' /root/namen
Klaus
klaus
root@ubuntu2004-101:~# grep '[kK][la]' /root/namen
Klaus
klaus
root@ubuntu2004-101:~# echo "karin" >> /root/namen
```

```
root@ubuntu2004-101:~# grep '[kK][la]' /root/namen
Klaus
klaus
karin
```

```
echo "Klaus1" >> /root/namen
root@ubuntu2004-101:~# echo "Klaus2" >> /root/namen
root@ubuntu2004-101:~# grep '[kK][la]aus[0-9]' /root/namen
```

Mengeangabe

```
## Achtung unbedingt egrep oder grep -E verwenden
cat /root/namen
AxB nix
AxB nix
abc nix
a nix

egrep '^[a-zA-Z]{1,3} nix' /root/namen
```

```
echo "ab nix" >> /root/namen
## Mindestens 2 Zeichen
root@ubuntu2004-101:~# egrep '^[a-zA-Z]{2,} nix' /root/namen
AxB nix
AxB nix
abc nix
ab nix
```

Nach Zahlen Suchen

```
echo "12345 namen" >> /root/namen
grep "[[:digit:]]\{5\}" /root/namen
```

Alle Zeilen mit port beginnend mit 1 und dann beliebig viele Zahlen und dann /tcp

```
cat /etc/services | grep -E '\s1[0-9]+\s/tcp'
```

Cheatsheets

- <https://cheatography.com/tme520/cheat-sheets/grep-english/>

Ref:

- <https://www.cyberciti.biz/faq/grep-regular-expressions/>

Finden von files nach Kriterien - find

Suchen nach allen Vorkommen im Verzeichnis /etc mit 'ssh' im Namen

```
## Bitte immer einfach Hochkommas verwenden.
find /etc -iname 'ssh*'
## Wäre das gleiche wie.
find /etc -iname 'ssh*' -print
```

Suchen nach allen Vorkommen von mariadb

```
## iname - case insensitive
find / -iname '*mariadb*'
```

Simple find command

```
## find directories with specific name
find / -name tmpfiles.d -type d
find /etc -name 'ssh*' -type f
```

Suchen mit exec ausführen

```
find . type f -exec ls -la {} \;
```

Suchen und löschen

```
cd
mkdir foo
cd foo/
touch dateia dateib
find . -type f
## Achtung zwischen {} und \; ist ein Leerzeichen
find . -type f -exec rm {} \;
ls -la

## Alternativ
find . type f -exec rm {} \;
```

nach inum (inode number suchen)

```
find / -inum +524300 -inum -525000 -exec ls -lai {} \;
```

Logs/Loganalyse

Logfile beobachten

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

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

Dienste debuggen

Walkthrough

```
## Dienst startet nicht / nach Ausführen von systemctl restart wird Fehlermeldung ausgegeben
systemctl restart mariadb.service

## Schritt 1 : status -> was sagen die logs (letzte 10 Zeilen)
systemctl status mariadb.service

## Nicht fündig-> Schritt 2:
journalctl -xeu mariadb.service
```

```
## Nicht fündig -> Schritt 3:
## Spezifisches Log von Dienst suchen
## und evtl. LogLevel von Dienst hochsetzen
## z.B. bei mariadb (durch Internetrecherche herausfinden)
less /var/log/mysql/error.log

## Nicht fündig -> Schritt 5
## Allgemeines Log
## Debian/Ubuntu
/var/log/syslog
## Redhat/Centos & SLES (OpenSuSE)
/var/log/messages
```

Wie verfahren bei SystemV

Wie bei walkthrough aber ab Schritt 4

Find error in logs quickly

```
cd /var/log/mysql
## -i = case insensitive // egal ob gross- oder kleingeschrieben
cat error.log | grep -i error
```

Schweizer Taschenmesser der Suche

```
## Fehler ist gummitulpe - option - falsch in Konfigurationsdatei, aber wo ?
grep -r gummitulpe /etc
```

Rsyslog

Alle Logs an zentralen Log-Server schicken

```
/etc/rsyslog.conf
## udp
*. * @192.168.10.254:514
## tcp
*. * @@192.168.10.254:514

Ref: https://www.tecmint.com/setup-rsyslog-client-to-send-logs-to-rsyslog-server-in-centos-7/
```

Journal analysieren

Journal für eine bestimmte unit anzeigen

```
journalctl -u ssh.service
journalctl -u ssh.service -e --since "2022-07-05 08:00" --until "2022-07-05 09:00"
```

Show all boots

```
journalctl --list-boots
0 3c3cf780186642ae9741b3d3811e95da Tue 2020-11-24 14:29:44 CET
lines 1-1/1 (END)
```

Show boot log

```
journalctl -b
```

Journal persistent

- Normalerweise (auf den meisten Systemen), überlebt das Journal kein Reboot

```
## persistent setzen
## Achtung: in /etc/systemd/journald.conf muss Storage=auto gesetzt sein
## Dies ist auch der Default - Fall
## Achtung Achtung: Alle gezeigten Einträge mit # am Anfang sind die Default-Werte (in
journald.conf)
mkdir /var/log/journal
systemctl restart systemd-journal-flush.service
```

Restrict how much is logged / data

```
## in /etc/systemd/journald.conf
SystemMaxUse=1G
```

journalctl

```
## ubuntu
journalctl -u ssh
```

Show journalctl for specific Field

```
## Hilfreich Damit man die Felder
journalctl -o json-pretty
journalctl _PID=1 # show all entries for systemd - command startet as first program after
kernel is loaded
journalctl _UID=120 # show all log entries for specific user
```

Welche Hilfe gibt es ?

```
man journald.conf
man journalctl
man systemd.journal-fields
```

Bash: Variablen, Kommandosubstitution und Quotes

Kommandos ausgeben und weiter verwenden

```
## Beispiele
echo $(date)
echo "Heute ist: "$(date)
DATUM=$(date)
echo "Das Datum ist: "$DATUM
```

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

```

Quoting

Beispiel 1:

```

DATUM=$(date)
root@ubuntu2004:~# echo "Das ist das heutige $DATUM"
Das ist das heutige Do 19. Mai 13:50:07 CEST 2022
root@ubuntu2004:~# echo 'Das ist das heutige $DATUM'
Das ist das heutige $DATUM

## Verschiedene Quotes nacheinander
echo "Das ist's:"'"Das ist das heutige $DATUM'

```

Mehrzeiliges Beispiel

```

Mehrzeiliges Beispiel
echo 'hallo
> das geht so
> .. oder auch icht'
hallo
das geht so
.. oder auch icht

```

Hochkommas verwenden

```

## Einfache gehen nur ausserhalb
echo 'Test'\'' so geht es weiter'
## oder mit $
echo '$Test\' so geht es weiter'

```

```
echo "\"Ein Sprichwort\""  
## oder  
echo '"Ein Sprichtwort'"
```

Die history

history aufrufen

```
history
```

Befehl aus der historie ausführen

```
!23 # der befehl 23 wird direkt  
## Achtung wird direkt ausgeführt ohne Nachfrage
```

STRG + r -> suche

suchstring eingeben, und er zeigt Eintrag aus der history, den man direkt mit Return ausführen kann.

Dienste/Runlevel(Targets verwalten)

Systemd Überblick

Units

```
*.target  
*.service  
*.timer
```

Targets

```
multi-user.target (ehemals runlevel 3: Multi User mit Netzwerk)  
graphical.target (ehemals runlevel 5: genau wie 3 mit grafischer Oberfläche)  
  
### Die wichtigsten systemctl/service  
  
### systemctl Beispiele
```

Status eines Dienstes überprüfen

```
service sshd status systemctl status sshd
```

Wie heisst der Dienst / welche Dienste gibt es ? (nur wenn der service aktiviert ist).

```
systemctl list-units -t service
```

für apache

```
systemctl list-units -t service | grep apache
```

die Abkürzung

```
systemctl -t service | grep apache
```

```
systemctl list-unit-files -t service | grep ssh
```

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
```

```
### Wie sehe ich, wie ein Service konfiguriert ist / Dienstekonfiguration anzeigen ?
```

z.B. für Apache2

```
systemctl cat apache2.service
```

```
### Wie kann ich rausfinden, wie die runlevel als targets heissen ?
```

```
cd /lib/systemd/system root@ubuntu2004-104:/lib/systemd/system# ls -la run*target lrwxrwxrwx 1 root root 15 Jan 6 20:47
runlevel0.target -> poweroff.target lrwxrwxrwx 1 root root 13 Jan 6 20:47 runlevel1.target -> rescue.target lrwxrwxrwx 1 root root
17 Jan 6 20:47 runlevel2.target -> multi-user.target lrwxrwxrwx 1 root root 17 Jan 6 20:47 runlevel3.target -> multi-user.target
lrwxrwxrwx 1 root root 17 Jan 6 20:47 runlevel4.target -> multi-user.target lrwxrwxrwx 1 root root 16 Jan 6 20:47 runlevel5.target -
> graphical.target lrwxrwxrwx 1 root root 13 Jan 6 20:47 runlevel6.target -> reboot.target
```

```
### Welche Dienste sind aktiviert/deaktiviert
```

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



```
### Dienste bearbeiten
```

```
systemctl edit sshd.service
```

Dann eintragen

```
[Unit] Description=Jochen's ssh-server
```

Dann speichern und schliessen (Editor)

nur falls es nicht funktioniert !

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

```
### Dienste finetunen (von zu Fall zu Fall möglich)
```

Restart=on-failure RestartSec=5 StartLimitInterval=400 StartLimitBurst=3

<https://unix.stackexchange.com/questions/507911/systemctl-what-is-the-meaning-of-restart-always>

```
### systemctl Cheatsheet
```

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

```
### journalctl
```

journalctl -u apache2.service

kurz. nur bei services

journalctl -u apache2

fortwährend

journalctl -u apache2 -f

json pretty ausgabe

hilft beim identifizieren von feldern

journalctl -u apache2.service -o json-pretty

```
### Script mit systemd verwalten und EnvironmentVariablen
```

```
### Schritt 1: Script erstellen
```

```
#!/bin/bash
```

vi /usr/local/bin/script-ng.sh

```
echo "script script-ng schreibt was ins log..." env date >> /var/log/script-ng.sh env >> /var/log/script-ng.sh
```

```
chmod u+x /usr/local/bin/script-ng.sh script-ng.sh
```

Sichtprüfung im Log

```
cat /var/log/script-ng.sh
```

```
### Schritt 2: service ohne env erstellen.
```

--force weil datei nicht existiert.

--full eine vollständige service-datei und nicht überschwierig

```
systemctl edit --full --force script.service
```

```
[Unit] Description=script von jochen
```

```
[Service] Type=oneshot ExecStart=/usr/local/bin/script-ng.sh RemainAfterExit=true StandardOutput=journal
```

```
[Install] WantedBy=multi-user.target
```

editor speichern und schliessen

```
### Schritt 3: Testen
```

```
systemctl status script systemctl start script systemctl status script
```

```
### Schritt 4: Datei /etc/default/script mit env-variablen anlegen
```

vi /etc/default/script

```
SCRIPT_VERSION=1.0 SCRIPT_MENU=simple
```

```
### Schritt 5: Service im Bereich [Service] wie folgt modifizieren
```

systemctl edit --full script.sh

...

```
[Service] Type=oneshot ExecStart=/usr/local/bin/script-ng.sh RemainAfterExit=true StandardOutput=journal
```

```
EnvironmentFile=/etc/default/script Environment="SPEED=fast" Environment="DRINKS=all"
```

```
### Schritt 6: Restart
```

```
systemctl restart script.service
```

Sichtprüfung im Journal

```
systemctl status script
```

in den logs

cat /var/log/script-ng.log

```
### Referenz:

* https://gist.github.com/drmalex07/d006f12914b21198ee43

### Systemd Service endless loop

### Walkthrough
```

#!/bin/bash

vi /usr/local/bin/loop.sh

```
while /bin/true do date echo "neuer Durchlauf" sleep 5 done
```

```
chmod u+x loop.sh
```

systemctl edit --force --full loop.service

```
[Unit] Description=script von jochen
```

```
[Service] Type=simple ExecStart=/usr/local/bin/loop.sh
```

```
[Install] WantedBy=multi-user.target
```

```
systemctl status loop systemctl start loop systemctl status loop
```

Evtl. aktivieren für nächsten reboot

```
### Systemctl - timers

### Show all timers
```

alle Timer anzeigen

```
systemctl list-timers
```

```
### How ?

* .timer and .service file next to each other

### Example ?
```

timer - file

```
root@ubuntu2004-104:/etc# systemctl cat systemd-tmpfiles-clean.timer
```

/lib/systemd/system/systemd-tmpfiles-clean.timer

SPDX-License-Identifier: LGPL-2.1+

This file is part of systemd.

systemd is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

```
[Unit] Description=Daily Cleanup of Temporary Directories Documentation=man:tmpfiles.d(5) man:systemd-tmpfiles(8)
```

```
[Timer] OnBootSec=15min OnUnitActiveSec=1d
```

Service - file

```
root@ubuntu2004-104:/etc# systemctl cat systemd-tmpfiles-clean.service
```

/lib/systemd/system/systemd-tmpfiles-clean.service

SPDX-License-Identifier: LGPL-2.1+

This file is part of systemd.

systemd is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

```
[Unit] Description=Cleanup of Temporary Directories Documentation=man:tmpfiles.d(5) man:systemd-tmpfiles(8)
```

```
DefaultDependencies=no Conflicts=shutdown.target After=local-fs.target time-set.target Before=shutdown.target
```

```
[Service] Type=oneshot ExecStart=systemd-tmpfiles --clean SuccessExitStatus=DATAERR IOSchedulingClass=idle
```

```
### Example Reference
```

```
* https://www.tutorialdocs.com/article/systemd-timer-tutorial.html
```

```
### Personal Timer (timer for user)

* https://nielsk.micro.blog/2015/11/11/creating-systemd-timers.html

### systemctl - timers - Übung

### Schritt 1: script erstellen und testen
```

```
#!/bin/bash
```

vi /usr/local/bin/scriptv2.sh

```
LOGTO=/var/log/scriptv2.log echo "script script-ng schreibt was ins log...." env date >> $LOGTO env >> $LOGTO
```

```
chmod u+x /usr/local/bin/scriptv2.sh scriptv2.sh
```

```
### Schritt 2: Service erstellen und testen
```

systemctl edit --force --full scriptv2.service

```
[Unit] Description=simple script for testing timer [Service] Type=oneshot ExecStart=/usr/local/bin/scriptv2.sh
##RemainAfterExit=true StandardOutput=journal
```

```
systemctl status scriptv2 systemctl start scriptv2 systemctl status scriptv2
```

```
### Schritt 3: Timer erstellen und testen
```

systemctl edit --force --full scriptv2.timer

```
[Unit] Description=Timer for scriptv2 [Timer] OnCalendar=*:0/5
```

```
[Install] WantedBy=basic.target
```

```
systemctl enable scriptv2.timer
```

```
systemctl status scriptv2.timer systemctl start scriptv2.timer systemctl status scriptv2.timer
```

```
systemctl list-timers
```

```
### Gegenüberstellung service etc/init.d/ systemctl
```

SySV

```
a) /etc/init.d/rsyslog status /etc/init.d/rsyslog start /etc/init.d/rsyslog status
```

b) service rsyslog

Systemd

geht auch (unter der Haube wird systemctl verwendet)

service rsyslog status

```
## Administration

### Dienste debuggen

### Walkthrough
```

Dienst startet nicht / nach Ausführen von systemctl restart wird Fehlermeldung ausgegeben

systemctl restart mariadb.service

Schritt 1 : status -> was sagen die logs (letzte 10 Zeilen)

systemctl status mariadb.service

Nicht fündig-> Schritt 2:

journalctl -xeu mariadb.service

Nicht fündig -> Schritt 3:

Spezifisches Log von Dienst suchen

und evtl. LogLevel von Dienst hochsetzen

z.B. bei mariadb (durch Internetrecherche herausfinden)

less /var/log/mysql/error.log

Nicht fündig -> Schritt 5

Allgemeines Log

Debian/Ubuntu

/var/log/syslog

REdhat/Centos & SLES (OpenSuSE)

/var/log/messages

```
### Wie verfahren bei SystemV
```

Wie bei walkthrough aber ab Schritt 4

```
### Find error in logs quickly
```

```
cd /var/log/mysql
```

-i = case insensitive // egal ob gross- oder kleingeschrieben

```
cat error.log | grep -i error
```

```
### Schweizer Taschenmesser der Suche
```

Fehler ist gummitulpe - option - falsch in Konfigurationsdatei, aber wo ?

```
grep -r gummitulpe /etc
```

```
### Neue Partition mit lvm
```

```
### keine Befehle lvdisplay, vgdisplay etc.
```

```
apt install lvm2
```

```
### Schritt 1: Partitionen vorbereiten
```

parted /dev/sda

partitionen erstellen z.B. 1 und 2

2x mkpart

und ein flag für lvm setzen

```
(parted) set 1 lvm on (parted) set 2 lvm on
```

```
quit
```

```
### Schritt 2: Physical Volumes vorbereiten (1. lvm Schritt)
```

```
pvcreate /dev/sda1 /dev/sda2 pvdisplay
```

```
### Schritt 3: Volumen Group erstellen (vg)
```

```
vgcreate vg0 /dev/sda1 /dev/sda2 vgdisplay
```

```
### Schritt 4: Logical Volume erstellen (lv)
```



```
lvcreate -n data -L500M vg0 lvdisplay
```

```
### Schritt 5: filesystem aufbringen (ext4) und probelhalber mounten
```

```
mkfs.ext4 /dev/vg0/data mkdir -p /mnt/platte mount /dev/vg0/data /mnt/platte
```

```
### Schritt 6: /etc/fstab
```

```
umount /mnt/platte
```

vi /etc/fstab

```
/dev/vg0/data /mnt/platte ext4 defaults 0 1
```

```
mount -av
```

```
reboot
```

teste, ist platte wieder eingehängt

```
### Schritt 7: Vergrößern des Logical Volumes
```

Voraussetzung, ausreichend Speicher in der volume group

oder dazufügen

1. neue partition erstellen (auch auf komplett neuer Platte möglich)

2. pvcreate /dev/sdb1

3. vgextend vg /dev/sdb1

--resizefs

```
sudo lvresize -L +200M --resizefs vg0/data
```

```
### Fall 2: logical volume erweitern
```

gleich mit Erweiterung des Filesystems

```
lvextend --resizefs -l +100%FREE /dev/mapper/ubuntu--vg-ubuntu--lv
```

```
### Reference:
```

```
* lvresize: https://www.digitalocean.com/community/tutorials/how-to-use-lvm-to-manage-storage-devices-on-ubuntu-18-04
```

```
### Verzeichnisse backup/restore mit tar
```

```
### Sichern / Backup
```

```
cd /usr/src tar cfvz _etc.20220617.tar.gz /etc tar tf _etc.20220617.tar.gz
```

```
### Entpacken (Vorbereitung)
```

```
mkdir foo mv _etc.20220617.tar.gz foo cd foo
```

```
### Entpacken (Variante 1)
```

```
tar xvf _etc.20220617.tar.gz
```

Aufräumen

```
rm -fR etc/
```

```
### Entpacken (Variante 2)
```

```
tar tf _etc.20220617.tar.gz
```

Achtung Fehler - weil falscher Pfad

```
tar xvf _etc.20220617.tar.gz etc/sysctl.d/99-sysctl.conf /etc/services echo $?
```

So geht's

```
tar xvf _etc.20220617.tar.gz etc/sysctl.d/99-sysctl.conf etc/services ls -la
```

```
### Referenz:  
  
* https://linuxconfig.org/how-to-create-incremental-and-differential-backups-with-tar  
  
## Apache  
  
### SSL - LetsEncrypt mit Ubuntu 22.04  
  
### Step 1:
```

prerequisite - apache runs already

```
sudo apt update sudo apt install -y certbot python3-certbot-apache
```

```
### Step 2: checking virtualhost configuration
```

```
sudo nano /etc/apache2/sites-available/projekt1-training-local.conf
```

```
### Attention: virtual host - domain must be different than hostname
```

e.g. ap1.t3isp.de (virtual host domain) != hostname

if this is not the case change hostname

```
hostnamectl set-hostname main.training.local
```

be sure to restart apache to take

```
systemctl restart httpd
```

```
### Step 3: Use certbot to configure
```

```
certbot --apache --register-unsafely-without-email
```

```
### Test with your browser
```

<https://ap1.t3isp.de>

```
### Test Certificate with ssl labs
```

```
* https://ssllabs.com
```

```
### Refs:
```

```
* https://www.digitalocean.com/community/tutorials/how-to-secure-apache-with-let-s-encrypt-on-ubuntu-22-04
```

```
## 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 mklable 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
```

Add-on: Eingehängte Partitionen anzeigen

```
mount
```

Aushängen

```
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
```

Nach dem Rebooten

```
mount | grep platte # taucht platte hier auf ?
```

```
## Boot-Prozess und Kernel

### Hintergründe zum Starten des Systems

### Reihenfolge (Phase 1)
```

1. BIOS-Selbsttest
2. MBR (512 Bytes auf der Festplatte) -> Bootloader Teil 1.
3. Bootmanager (Grub) wird gestartet
4. Im Boot-Manager -> Auswahl welcher Eintrag soll verwendet werden
5. Mutter aller Prozesse -> init -> systemd

```
### Reihenfolge (Phase 2) - systemd hat die Kontrolle
```

in welchen Endmodus -> Target soll ich mich hocharbeiten

graphical.target

graphical.target (isolateTarget) multi-user.target (isolatedTarget) basic.target

```
### 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 Params beim Booten übergeben
```

```
### Welche kann ich in Ubuntu übergeben
```

Im Bootmanager (grub) -> Menüpunkt -> e - Taste drücken -> bei Zeile linux ganz am Ende

man kernel-command-line

Danach F10 oder CTRL-x

```
### 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
```

```
### -h oder --help --> eines geht immer
```

Beispiel ls

```
ls -h # geht nicht für Hilfe ls --help # geht !
```

```
### Navigation in den man-pages
```

q - verlassen von man Pfeil oben/unten PageUp/PageDown G # für ans Ende der Datei springe 1g # in die erste Zeile

```
### Suche mit in man-pages
```

/Suchwort [Enter] n # nächster Treffer (kleines n) N # letzter Treffer

```
## Grafische Oberfläche und Installation
```

```
### Gnome unter Ubuntu installieren
```

sudo apt install taskel sudo taskel install ubuntu-desktop

```
### X-Server - Ausgabe auf Windows umleiten
```

```
* https://www.thomas-krenn.com/de/wiki/Grafische_Linux_Programme_remote_von_einem_Windows_PC_mit_Xming_nutzen
```

```
### Installations-Images-Server
```

```
* https://ubuntu.com/download/server#download
```

```
## Wartung, Sicherung und Aktualisierung
```

```
### Aktualisierung des Systems
```

apt update apt upgrade apt dist-upgrade

oder geht auch auf älteren Systemen

apt-get update apt-get upgrade apt-get dist-upgrade

```
### Paketmanager apt/dpkg
```

```
### Alle Pakete anzeigen, die installiert sind auf dem System
```

dpkg -l

oder

apt list --installed

```
### Alle Paket die zur Verfügung stehen
```

apt list

```
### Wo sind die Repos konfiguriert
```

cat /etc/apt/sources.list cd /etc/apt/sources.list.d

```
### Paket deinstallieren und aufräumen
```

mit Konfigurationsdateien deinstallieren

```
apt purge mariadb-server
```

Konfigurationsdateien stehen lassen

```
apt remove mariadb-server
```

Aufräumen / alle Pakete die nicht mehr benötigt werden

```
apt autoremove
```

```
### Pakete händisch mit dpkg installieren
```

Schritt 1: Im Browser

Paket online finden und Link kopieren (Browser - Rechte Mausataste Link kopieren)

Schritt 2: auf dem Linux Server

```
sudo apt install wget cd /usr/src wget http://archive.ubuntu.com/ubuntu/pool/main/a/acl/acl\_2.2.53-10build1\_amd64.deb sudo  
dpkg -i acl_2.2.53-10build1_amd64.deb
```

```
### Pakete mit apt search suchen
```

Vorbereitung

```
apt update
```

suche nache apache

```
apt search apache
```

mit pager

```
apt search apache | less
```

Alle Paket in denen apache am Anfang der Zeile fehlt

```
apt search ^apache | less
```

```
### Installieren mit apt install
```

mit genauem Namen

apt install apache2

```
### Liste der Files aus dem Paket (wenn installiert)
```

dpkg -L openssh-server

```
### Paket runterladen, wenn bereits installiert
```

apt install -d --reinstall openssh-server # -d steht für download-only

Lädt das Paket unter

/var/cache/apt/archives runter

```
### Welche Dateien sind im Paket, die ausgerollt werden ? (ohne Installation)
```

cd /var/cache/apt/archives dpkg --contents openssh-server-xyz.deb # im gleichen Verzeichnis oder vollen Pfad dorthin

oder Paket haben händisch in ein anderes Verzeichnis runtergeladen (z.B. mit wget)

dpkg -c /usr/src/openssh-server-xyz.deb

```
### Archive runterladen und entpacken
```

Walkthrough

Schritt 1: Download-Link in Browser kopieren (rechte Maustaste)

Schritt 2:

cd /usr/src

falsche Dateiname -> umbenannt.

wget <https://github.com/phayes/geoPHP/tarball/master> mv master master.tar.gz

Schritt 3: Sicherheitsverzeichnis anlegen und entpacken

mkdir foo mv master.tar.gz foo cd foo tar xvf master.tar.gz

```
### Mehrere Versionen eines Programms z.B. php (cli) verwalten
```

```
### Ref:
```

```
* https://devanswers.co/run-multiple-php-versions-on-apache/
```

```
### Verzeichnisse in archiven sichern - tar
```

```
### Sichern
```

```
tar cvfz /usr/src/_etc.20220522.tar.gz /etc
```

```
## Liste der Dateien aus Archiv anzeigen
```

```
tar tf /usr/src/_etc.20220522.tar.gz
```

Gibt es die Datei ?

```
tar tf /usr/src/_etc.20220522.tar.gz /etc/skel/.bashrc
```

```
## Dateien aus Archiv entpacken
```

```
mkdir auspackverzeichnis cp -a _etc.20220522.tar.gz auspackverzeichnis cd auspackverzeichnis tar xvf  
/usr/src/_etc.20220522.tar.gz
```

Einzelne Dateien

```
tar xvf /usr/src/_etc.20220522.tar.gz etc/skel/.bashrc
```

```
## Absicherung System
```

```
### ssh absichern
```

```
### Walkthrough
```

Schritt 1:

nano /etc/ssh/sshd_config

ganz unten hinzufügen

AllowGroups sshadmin

```
``
```

```
## wichtig !! auch der root-Benutzer, muss dann der Gruppe angehören  
## sollte ich mich mit dem Root-Benutzer über public/private key verbinden
```

Schritt 2:

Jeden Benutzer der sich mit ssh verbinden können soll, der Gruppe sshadmin hinzufügen

```
usermod -aG sshadmin kurs
```

Schritt 3:

```
systemctl restart sshd
```

```
### Achtung: /etc/ssh/sshd_config.d
```

Diese Dateien überschreiben die config aus /etc/ssh/sshd_config

```
## Firewall und ports

### ufw (uncomplicated firewall)

### Läuft der Dienst für die Firewall
```

```
systemctl status ufw
```

```
### Ist die Firewall scharfgeschaltet ?
```

```
ufw status
```

```
### Firewall aktivieren (Achtung ssh)
```

Neue ssh - Verbindungen werden nicht angenommen

Bestehende Bedingungen (ESTABLISHED) bleiben erhalten

```
ufw enable ufw status
```

```
### Port hinzufügen
```

```
ufw allow 22 # for tcp and udp
```

or

```
ufw allow ssh # uses /etc/services for detection of port - number ufw status
```

```
### Port wieder rausnehmen
```

```
ufw delete allow http ufw delete allow ssh ufw delete allow 22
```

ufw status numbered

e.g.

ufw delete 1

```
### firewallld

### Install firewallld and restrict ufw
```

Schritt 1: ufw deaktivieren

systemctl stop ufw systemctl disable ufw ufw disable # zur Sicherheit ufw status

-> disabled # this has to be the case

Schritt 2: firewallld

apt install firewallld systemctl status firewallld

doublecheck ufw

systemctl status ufw

```
### Is firewallld running ?
```

is it set to enabled ?

systemctl status firewallld firewall-cmd --state

```
### Command to control firewallld

* firewall-cmd

### Zones documentation

man firewallld.zones

### Zones available
```

firewall-cmd --get-zones block dmz drop external home internal public trusted work

```
### Active Zones
```

```
firewall-cmd --get-active-zones
```

in our case empty

```
### Add Interface to Zone = Active Zone
```

```
firewall-cmd --zone=public --add-interface=enp0s3 firewall-cmd --runtime-to-permanent firewall-cmd --get-active-zones
```

public

interfaces: enp0s3

```
### Show information about all zones that are used
```

```
firewall-cmd --list-all firewall-cmd --list-all-zones
```

```
### Default Zone
```

if not specifically mentioned when using firewall-cmd

.. add things to this zone

```
firewall-cmd --get-default-zone public
```

```
### Show services / Info
```

```
firewall-cmd --get-services firewall-cmd --info-service=http
```

```
### Adding/Removing a service
```

Version 1 - more practical

set in runtime

```
firewall-cmd --zone=public --add-service=http firewall-cmd --runtime-to-permanent
```

Version 2 - less practical

```
firewall-cmd --permanent --zone=public --add-service=http firewall-cmd --reload
```

Service wieder entfernen

```
firewall-cmd --permanent --zone=public --remove-service=ssh firewall-cmd --reload
```

```
### Best way to add a new rule
```

Step1: do it persistent -> written to disk

```
firewall-cmd --add-port=82/tcp --permanent
```

Step 2: + reload firewall

```
firewall-cmd --reload
```

```
### Enable / Disabled icmp
```

```
firewall-cmd --get-icmptypes
```

none present yet

```
firewall-cmd --zone=public --add-icmp-block-inversion --permanent firewall-cmd --reload
```

```
### Working with rich rules
```

Documentation

man firewalld.richlanguage

throttle connectons

```
firewall-cmd --permanent --zone=public --add-rich-rule='rule family=ipv4 source address=10.0.50.10/32 service name=http log level=notice prefix="firewalld rich rule INFO: " limit value="100/h" accept' firewall-cmd --reload # firewall-cmd --zone=public --list-all
```

port forwarding

```
firewall-cmd --get-active-zones firewall-cmd --zone=public --list-all firewall-cmd --permanent --zone=public --add-rich-rule='rule family=ipv4 source address=10.0.50.10 forward-port port=42343 protocol=tcp to-port=22' firewall-cmd --reload firewall-cmd --zone=public --list-all firewall-cmd --remove-service=ssh --zone=public
```

list only the rich rules

```
firewall-cmd --zone=public --list-rich-rules
```

persist all runtime rules

```
firewall-cmd --runtime-to-permanent
```

```
### References
```

```
* https://www.linuxjournal.com/content/understanding-firewalld-multi-zone-configurations#:~:text=Going%20line%20by%20line%20through,or%20source%20associated%20with%20it.
```

```
* https://www.answertopia.com/ubuntu/basic-ubuntu-firewall-configuration-with-firewalld/

### Scannen und Überprüfen mit telnet/nmap

## Netzwerk/Dienste

### Wie Netzwerk einrichten in unterschiedlichen Distros

### Debian
```

/etc/network/interfaces (ifupdown - package)

```
### Ubuntu

#### Desktop
```

netplan -> NetworkManager

Zum Konfigurieren

nmtui nmcli

```
### Server
```

netplan -> networkd /etc/netplan netplan - Kommando

```
### SLES 15
```

wicked yast

```
### Redhat / Centos / Rocky
```

NetworkManager

Tools

nmtui nmcli

```
### Hostname setzen
```

please do it root

hostnamectl hostnamectl set-hostname server1.training.local

only reflects after new login

su -

```
### netplan unter Ubuntu
```

```
### Desktop
```

/etc/netplan/01-network-manager-all.yaml

On Desktop-Systems everything is handled by the Network-Manager

Let NetworkManager manage all devices on this system

```
network: version: 2 renderer: NetworkManager
```

```
### Umschalten auf anderen Renderer
```

...yaml

```
network: version: 2 renderer: networkd ethernets: enp0s3: dhcp4: true
```

check changes

```
netplan try
```

apply changes

```
netplan apply
```

```
### Statische Adresse konfigurieren
```

routing rausfinden

```
ip route
```

Verwendete Nameserver rausfinden

```
cat /etc/resolv.conf
```

ergänzen

```
enp0s8: addresses: - 192.168.56.101/24 routes: - to: default via: 10.0.2.2 nameservers: search: [training.local] addresses: [8.8.8.8]
```

```
netplan apply --debug
```


or

netplan apply

or

netplan try

```
### Interface händisch (nicht persistent) stoppen
```

ip link set dev enp0s8 down

```
### Alternative: ifupdown - package als Alternative

* Empfehlung: Wir bleiben dabei, was das System mir anbieten (also netplan)
* ifupdown bietet die klassischen /etc/network/interfaces zur Konfiguration

### Referenz

* https://netplan.io/examples/

### Netzwerk unter SLES - OpenSUSE - wicked

### Dienste

#### Netzwerk-Client starten
```

systemctl status network

oder

systemctl status wicked

```
### Wicked-Daemon
```

systemctl status wickedd.service

```
### Welche Dienste gibt es ?
```

systemctl list-units -t service | grep wicked

```
### Journal abrufen
```

journalctl -u wicked.service

```
### wicked - commands
```

wicked show all wicked show lo wicked show eth0

wicked show-xml all

```
### Wicked Netzwerk-Karte hochziehen/runterziehen
```

wicked ifdown eth0 wicked show eth0 wicked ifup eth0 wicked show eth0

```
### Auf welchen Ports lauscht mein Server - lsof
```

Zeigt alle ports an auf die gelauscht wird (ipv4)

lsof -i

alternative

netstat -tupel

```
## Bash/Bash-Scripting
```

```
### Einfaches Script zur Datumsausgabe
```

```
### Beliebiges Verzeichnis
```

Mit nano öffnen / datei muss vorher nicht vorhanden sein

nano script.sh

Folgendes muss drin stehen, mit 1. Zeile beginnend mit

```
#!/bin/bash date
```

Speichern CTRL + O -> RETURN, CTRL X

Ausführbar machen

chmod u+x script.sh ./script.sh # Ausführen und wohlfühlen

```
### Besser: /usr/local/bin
```

```
#### Gründe
```

- * Wird gefunden, egal in welchem Verzeichnis ich bin ?
- * Warum ? Weil /usr/local/bin Teil der PATH-Variablen ist

```
#### Beispiel
```

```
cd /usr/local/bin
```

Mit nano öffnen / datei muss vorher nicht vorhanden sein

nano script.sh

Folgendes muss drin stehen, mit 1. Zeile beginnend mit

```
#!/bin/bash date
```

Speichern CTRL + O -> RETURN, CTRL X

Ausführbar machen

```
chmod u+x script.sh
```

z.B.

```
cd /etc
```

Script wird ausgeführt, weil die bash es im Verzeichnis /usr/local/bin findet

anhand von \$PATH

```
echo $PATH env script.sh
```

```
### Script Beispiel
```

```
### Beispiel 1: for
```

```
cd /usr/local/bin vi liste.sh
```

```
#!/bin/bash
```

```
LISTE="etc dev boot" for i in $LISTE do ls -la "$i" done
```

```
chmod o+x liste.sh
```

Testen

```
liste.sh
```

```
### Beispiel 2: for mit befehl und Konfig-Datei
```

vi /etc/liste.conf

```
LOGTO=/var/log/liste.log
```

vi /usr/local/bin/liste.sh

```
#!/bin/bash
```

DATUM=\$(date)

```
source /etc/liste.conf echo $LOGTO
```

```
LISTE=$(echo $PATH | tr ':' ' ')
```

```
for i in $LISTE do ls -la "$i" >> $LOGTO done
```

```
chmod o+x /usr/local/bin/liste.sh
```

testen

liste.sh

```
### Beispiel 3: mit Übergabe parameter
```

```
vi /usr/local/bin/dirmaker.sh
```

```
#!/bin/bash
```

```
##echo $0 ##echo $1
```

echo \$#

```
if test "$1" = "" then read -p "Verzeichnisname?" VERZ else VERZ=$1 echo $VERZ fi
```

```
if test ! -d /tmp/$VERZ then echo "Verzeichnis /tmp/$VERZ existiert nicht" echo "..." echo "... wird angelegt" mkdir /tmp/$VERZ
```

```
else echo "Verzeichnis /tmp/$VERZ existiert. Punkt !" ls -la /tmp/$VERZ fi
```

Ausführen/Verketteten von mehreren Befehlen

```
## Beide Befehle ausführen, auch wenn der 1. fehlschlägt  
befehl1; apt upgrade
```

```
## 2. Befehl nur ausführen, wenn 1. erfolgreich war.  
apt update && apt upgrade
```

```
## 2. Befehl nur ausführen, wenn der 1. NICHT erfolgreich war  
## befehl1 oder befehl2 (im weitesten Sinne)  
befehl1 || befehl2
```

Vordefinierte Variablen z.B \$0

```
#!/bin/bash
echo "0:"$0
echo "1:"$1
echo "2:"$2
echo "@:"$@
echo '#'$#
echo Hallo heute ist:$(date)

##exit 22
```

Funktionen in der bash

```
#!/bin/bash
## vi /usr/local/bin/functiontest.sh
## chmod u+x /usr/local/bin/functiontest.sh

LOGTO=/var/log/logme

function logto {
    # echo "hello jochen"
    date >> $LOGTO
    echo "hello jochen: $1" >> $LOGTO
}

logto 'Hans hat Glück'
echo "----"
cat $LOGTO
```

```
## im script wird Funktion aufgerufen
functiontest.sh
```

Best practice structure bash - scripts

File 1: config.sh

```
## vi /usr/local/bin/config.sh
LOGTO=/var/log/logme
DATUM=$(date)
```

File 2: functions.sh

```
## vi /usr/local/bin/functions.sh
function logto {
    # echo "hello jochen"
    date >> $LOGTO
    echo "hello jochen: $1" >> $LOGTO
}
```

File 3: script.sh

```
## vi /usr/local/bin/script.sh
##!/bin/bash

source /usr/local/bin/config.sh
source /usr/local/bin/functions.sh
```

```
logto 'Hans hat Glück'
echo "----"
cat $LOGTO
echo $LOGTO
echo $DATUM
```

```
chmod u+x script.sh
script.sh
```

Neue Umgebungsvariable setzen

example

```
sudo su -
cd
## root-verzeichnis /root
pwd
echo "export NACHNAME=Mustermann" >> .bashrc

### testen - neu als root einloggen
su -
## Jetzt müsste NACHNAME in den Umgebungsvariablen zu sehen sein
env
```

Timers/cronjobs

Cronjob - hourly einrichten

Walkthrough

```
cd /etc/cron.hourly
## nano datum
## wichtig ohne Endung
## Job wird dann um 17 nach ausgeführt ?

####

##!/bin/bash
date >> /var/log/datum.log

chmod 755 datum # es müssen x-Rechte (Ausführungsrechte gesetzt sein)

### Abwarten, Tee trinken
```

cronjob (zentral) - crond

```

cd /etc/cron.d
### cronjob anlegen
## Achtung: ohne Dateiendung
## ls -la trainingscript
## root@ubuntu2004-104:/etc/cron.d# ls -la trainingscript
## -rw-r--r-- 1 root root 471 Mar 26 12:44 trainingscript

## cat trainingscript
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

## Example of job definition:
## .----- minute (0 - 59)
## | .----- hour (0 - 23)
## | | .----- day of month (1 - 31)
## | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
## | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
## | | | | |
## * * * * * user-name command to be executed
*/2 * * * * root /root/script.sh

### Script anlegen
cat script.sh
#!/bin/bash
TAG='FREITAG'
echo " ---- " >> /var/log/scripting.log
date >> /var/log/scripting.log
echo $TAG >> /var/log/scripting.log

### Script - Berechtigungen setzten
chmod u+x /root/script.sh

### Scriptausführung testen
### trägt es etwas im Log ein -> /var/log/scripting.log
/root/script.sh

##### Warten

### nach 2 Minuten log betrachten
ls -la /var/log/scripting.log

### cron daemon braucht nicht reloaded zu werden

```

Übungen

Übung Verzeichnis

1. verzeichnis planung anlegen
2. ins verzeichnis planung wechseln
3. rekursiv struktur anlegen: woche1/tag1/stunden

4. Leere datei anlegen : tag1
5. verzeichnisstruktur wieder löschen

Übung Dienste

1. Ihr sucht nach dem MariaDB-Server mit Hilfe von apt search
2. Ihr installiert den mariadb server mit dem Flag ohne Nachfrage -y
3. Ihr findet raus unter welchem Dienst der Server läuft (mariadb)
4. Ihr stoppt den Server
5. Ihr deaktiviert den (disable)
6. Ihr startet den Server
7. Ihr aktiviert den Server für das nächste Reboot
8. Lass Euch im letzten Schritt alle logs anzeigen

Übung Umleitung mit Variable

Übung user/password

Literatur

Literatur

Literatur

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

Cheatsheet

- [Cheatsheet bash](#)
- [..ansonsten Google :o\)](#)

Bash - Programmierung

- [Bash Programmierung](#)
- [Bash Advanced Programmierung](#)

Cheatsheet Commandline

- <https://cheatography.com/davechild/cheat-sheets/linux-command-line/pdf/>

Wo finde ich Hilfe im Internet

Einfache Fragen

```
Linux quoting (bei google)
In der Regel kommen wir dann auf Stackoverflow
https://stackoverflow.com/questions/6697753/difference-between-single-and-double-quotes-in-bash
```

Größere Fragen

```
Wie installiere ich
apache2 ubuntu howto
apache2 ubuntu 20.04 howto
apache2 ubuntu 20.04. DocumentRoot
```



```
https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-20-04-de
```

RDP-Client unter Windows

- <https://linuxwiki.de/rdesktop>

Linux Malware Scanner

- <https://www.rfxn.com/projects/linux-malware-detect/>

Erweiterte Befehle (Nice to have)

Welche Bibliotheken verwendet ein ausführbares Programm

```
ldd /usr/bin/ls
```

Tools/Verschiedens

Remote Desktop für Linux / durch Teilnehmer getestet

- <https://wiki.ubuntuusers.de/Remmina/>

Warum umask 002 und 0002 ? - Geschichte

```
## Just quoting redhat here.

The setting which determines what permissions are applied to a newly created file or
directory is called a umask and is configured in the /etc/bashrc file.

Traditionally on UNIX systems, the umask is set to 022, which allows only the user
who created the file or directory to make modifications. Under this scheme,
all other users, including members of the creator's group, are not allowed
to make any modifications. However, under the UPG scheme, this "group protection"
is not necessary since every user has their own private group.

## Ref:
https://access.redhat.com/documentation/en-
us/red_hat_enterprise_linux/4/html/reference_guide/s1-users-groups-private-groups
```

lokale Mails installieren

```
apt install postfix mailutils
## Internet Host

echo "testmail" | mail -s "subject" root

## Gucken in der Datei
cat /var/mail/root
## nach der gesendeten Email
```

Debian/Ubuntu - deb - Paket entpacken

```
apt install -y binutils
cd /usr/src
mkdir unwrap-folder
mv openssh-server-xyz.deb unwrap-folder
```

```
cd unwrap-folder
tar x openssh-server-xyz.deb
## additionally might be necessary
tar xvf data.tar.xz
tar xvf control.tar.xz
```

Geänderte Dateien zu anderem Server schicken

```
rsync --links -u -v -e ssh -r /var kurs@192.168.56.102:/home/kurs
```

Prozesse

Prioritäten und NiceNess

nice aus dem userspace möglich

```
## userspace - ich führe als Benutzer ein Programm
## nice - Festlegen ein programm zum Kernel ist

## Priorität in top - pr
## je niedriger die Zahl für den Prozess in top ist, desto
## höher die Priorität
## höchste Priorität -100
-100 bis 39

## -100 = rt
## Statt -100 steht rt dort in top in der Spalte PR

## -59 höhe Priorität als 0 als 20 etc.

## Aus nice - Anfragen. von -20 -> +19 machen der Kernel
## in der Regel
n + 20 z.B. Niceness= -20. = -20 + 20 = 0
## D.h. höchste Priorität über nice im Kernel ist 0
## die niedrigste Priorität 39
```

Prozesse in Realtime ausführen wie geht ?

```
chrt # change realtime
## Welche Warteschlangen gibt es
chrt -m

## Packe top in die realtime Warteschlange und führe es aus
## --rr realtime roundrobin Warteschlange
## Führe top mit der Priorität -100 aus
chrt --rr 99 top
```

Ref:

- <https://medium.com/@joseagustin.barra/understanding-priority-levels-in-linux-cd8c82eb4dd>

Netzwerk

IP-Adresse von DHCP-Server holen (quick-and-dirty)

Walkthrough

```
## Ip nicht gesetzt - kurzfristig eine IP holen
ip a # zeigt die Netzwerkschnittstellen an.
dhclient enp0s8 # ip - Adresse für Schnittstelle enp0s8 holen
ip a
```

Digitalocean

Script zum Aufsetzen eines Server mit Docker

cloud-init script

```
#!/bin/bash
groupadd sshadmin
USERS="11trainingdo"
for USER in $USERS
do
    echo "Adding user $USER"
    useradd -s /bin/bash $USER
    usermod -aG sshadmin $USER
    echo "$USER:hier-kommt-das-passwort-rein" | chpasswd
done

## We can sudo with 11trainingdo
usermod -aG sudo 11trainingdo

## Setup ssh stuff
sed -i "s/PasswordAuthentication no/PasswordAuthentication yes/g" /etc/ssh/sshd_config
usermod -aG sshadmin root
echo "AllowGroups sshadmin" >> /etc/ssh/sshd_config
systemctl reload sshd

## specifically fix do - stuff in new versions of cloud-init
sed -i "s/PasswordAuthentication no/PasswordAuthentication yes/g" /etc/ssh/sshd_config.d/50-
cloud-init.conf

apt update
## install dependencies
sudo apt install apt-transport-https curl gnupg-agent ca-certificates software-properties-
common -y

## get the gpgkey
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

## prepare repo for jammy (22.04 LTS)
add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu jammy stable"

## now install the packages
apt install docker-ce docker-ce-cli containerd.io -y
```