Siduction Handbuch

siduction Team

15.05.2022

Inhaltsverzeichnis

1 Willkommen **11**
1.1 Das siduction™ GNU/Linux-Betriebssystem . . . 11
1.1.1 Allgemeines . . . 11
1.1.2 Copyright Rechts- und Lizenzhinweise . . . 12
1.1.3 Haftungsausschluss . . . 12
1.2 Das siduction-Team . . . 13
1.2.1 Credit für siduction 2021.2.0 . . . 13
1.2.1.1 Core Team . . . 13
1.2.1.2 Art Team . . . 13
1.2.1.3 Code Ideen Unterstützung Handbuch . . . 13
1.2.1.4 Credit für das original manual Team . . . 13
1.3 siduction Hilfe . . . 15
1.3.1 Das siduction-Forum . . . 15
1.3.2 Das siduction-Wiki . . . 15
1.3.3 IRC - interaktiver Livesupport . . . 15
1.3.4 Nützliche Helfer im Textmodus . . . 16
1.3.5 siduction IRC-Support im Textmodus . . . 17
1.3.6 Surfen im Internet im Textmodus . . . 19
1.3.7 inxi . . . 20
1.3.8 Nützliche Links . . . 20

2 Quickstart **22**
2.1 siduction Kurzanleitung . . . 22
2.1.1 Essenzielle Kapitel . . . 22
2.1.2 Zur Stabilität von Debian Sid . . . 23
2.1.3 Der siduction-Kernel . . . 23
2.1.4 Die Verwaltung von Softwarepaketen . . . 23
2.1.5 Aktualisierung des Systems - upgrade . . . 24
2.1.6 Konfiguration von Netzwerken . . . 25
2.1.7 Runlevels - Ziel-Unit . . . 26
2.1.8 Weitere Desktopumgebungen . . . 26
2.1.9 Hilfe im IRC und im Forum . . . 26

3 ISO-Abbilder **27**
3.1 Inhalt der Live-ISO . . . 28

3.1.1 Hinweis zur Software auf dem Live-ISO 28
3.1.2 Varianten der ISO 28
3.1.3 Minimale Systemanforderungen 29
3.1.4 Anwendungen und Hilfsprogramme 30
3.1.5 Haftungsausschluss_Disclaimer 30
3.2 Live-DVD verwenden 32
3.2.1 Eingerichtete User auf dem Live-Medium 32
3.2.2 Mit root-Rechten auf der Live-DVD 32
3.2.3 Ein neues Passwort setzen 33
3.2.4 Software-Installation bei Live-Sitzung 33
3.3 Bootoptionen Cheatcodes 35
3.3.1 siduction spezifische Parameter 35
3.3.2 Bootoptionen für den Grafikserver X 37
3.3.3 Allgemeine Parameter des Linux-Kernels 39
3.3.4 VGA-Codes 40
3.4 ISO download 42
3.4.1 Dateien der siduction-Spiegelserver 44
3.4.2 Integritätsprüfung 45
3.5 ISO auf USB-Stick - Speicherkarte 47
3.5.1 GUI Anwendung 47
3.5.2 Linux Kommandozeile 48
3.5.2.1 Zusätzliche Datenpartition 49
3.5.3 Mac OS X Kommandozeile 50
3.6 ISO brennen 51
3.6.1 DVD mit Linux brennen 51
3.6.2 DVD mit Windows brennen 52
3.7 Life-DVD ohne GUI brennen 53
3.7.1 burniso 53
3.7.2 Brennen mit cdrdao wodim growisofs 54
3.7.3 Verfügbare Geräte 54
3.7.4 Beispiele für CD DVD BD 55

4 Installation **58**
4.1 Installation auf HDD 59
4.1.1 Datensicherung 59
4.1.2 Installationsvorbereitungen 59

4.1.3 Partitionierung 60
4.1.4 Dateisysteme 60
4.1.5 Duplizierung auf einen anderen Computer 61
4.1.6 Das Installationsprogramm Calamares 61
4.1.7 System verschlüsseln 66
4.1.8 Benutzer hinzufügen 70
4.2 Aus ISO-Datei booten 71
4.2.1 Überblick 71
4.2.2 fromiso mit Grub2 71
4.2.3 toram 73
4.3 Partitionierung von Installationsmedien 74
4.3.1 Mindestanforderungen 74
4.3.2 Beispiele mit verschiedenen Plattengrößen 75
4.3.3 Partitionierungsprogramme 77
4.3.4 Weiterführende Infos 78
4.4 UUID - Benennung von Blockgeräten 79
4.4.1 Arten der Benennung von Blockgeräten 79
4.4.2 Label verwenden 80
4.5 Die fstab 81
4.5.1 Anpassung der fstab 82
4.5.2 Erstellung neuer Einhängepunkte 83
4.6 Partitionieren mit GParted 85
4.6.1 Wichtige Hinweise 85
4.6.2 GParted verwenden 85
4.6.3 fstab anpassen 92
4.6.4 NTFS-Partitionsgrößen mit GParted ändern 94
4.7 Partitionieren mit gdisk 95
4.7.1 Partitionierung einer Festplatte 96
4.7.2 cgdisk verwenden 97
4.7.2.1 Partition erstellen 98
4.7.2.2 Partition löschen 102
4.7.2.3 GPT schreiben 102
4.7.3 Formatieren der Partitionen 103
4.7.4 Booten mit GPT-UEFI oder GPT-BIOS 104
4.7.4.1 Booten mit UEFI 105
4.7.4.2 Booten mit BIOS 105

4.7.5 Erweiterte Befehle von gdisk . . . 105
4.8 Partitionieren mit fdisk . . . 108
4.8.1 Benennung von Speichergeräten . . . 108
4.8.2 Cfdisk verwenden . . . 111
4.8.2.1 Die Bedienoberfläche . . . 111
4.8.2.2 Löschen einer Partition . . . 112
4.8.2.3 Größe einer Partition ändern . . . 112
4.8.2.4 Erstellen einer neuen Partition . . . 113
4.8.2.5 Partitionstyp . . . 115
4.8.2.6 Eine Partition bootfähig machen . . . 116
4.8.2.7 Partitionstabelle schreiben . . . 116
4.8.2.8 Cfdisk beenden . . . 117
4.8.3 Formatieren von Partitionen . . . 117
4.9 LVM-Partitionierung - Logical Volume Manager . . . 119
4.9.1 Sechs Schritte zu Logical Volumes . . . 120
4.9.2 Größenänderung eines Volumens . . . 122
4.9.3 LVM mit einem GUI-Programm verwalten . . . 124
4.9.4 Weitere Infos . . . 124
4.10 Das Verzeichnis home verschieben . . . 125
4.10.1 Private Daten verschieben . . . 126
4.10.2 fstab anpassen . . . 130

5 Netzwerk **131**
5.1 Network Manager Kommandline Tool . . . 132
5.1.1 Network Manager verwenden . . . 132
5.1.2 Weiterführende Informationen . . . 136
5.2 IWD . . . 137
5.2.1 Grafische Konfigurationsprogramme . . . 137
5.2.2 Konfiguration im Terminal . . . 137
5.3 IWD statt wpa_supplicant . . . 140
5.3.1 IWD installieren . . . 141
5.3.2 Zurück zum wpa_supplicant . . . 144
5.4 SAMBA . . . 145
5.4.1 Client-Konfiguration . . . 145
5.4.2 siduction als Samba-Server . . . 146
5.5 SSH . . . 147

5.5.1 SSH absichern . . . 147
5.5.2 SSH für X-Window Programme . . . 149
5.5.3 Kopieren scp via ssh . . . 150
5.5.4 SSH mit Dolphin . . . 152
5.5.5 SSHFS - auf einem entfernten Computer mounten . . . 153
5.6 LAMP-Webserver . . . 155
5.6.1 Apache installieren . . . 157
5.6.2 MariaDB installieren . . . 159
5.6.3 PHP installieren . . . 159
5.6.4 phpMyAdmin installieren . . . 161
5.6.5 Weitere Software . . . 162
5.6.6 Statusaugaben Log-Dateien . . . 162
5.6.7 Troubleshooting . . . 164
5.6.7.1 Wenn nichts hilft . . . 167
5.6.8 Sicherheit . . . 168
5.7 Apache einrichten . . . 169
5.7.1 Apache im Dateisystem . . . 169
5.7.2 Verbindung zum Server . . . 170
5.7.3 Apache Konfiguration . . . 171
5.7.4 Benutzer und Rechte . . . 173
5.7.5 Sicherheit - Apache Standard . . . 176
5.7.6 Sicherheit - weitere Konfigurationen . . . 177
5.7.7 HTTPS verwenden . . . 178
5.7.8 Sicherheits Tipps . . . 179
5.7.9 Integration in Apache2 . . . 179
5.7.10 Quellen Apache . . . 181
5.8 MariaDB einrichten . . . 182
5.8.1 MariaDB im Dateisystem . . . 182
5.8.2 Erstkonfiguration . . . 182
5.8.3 MariaDB CLI . . . 185
5.8.3.1 Eine Datenbank anlegen . . . 185
5.8.3.2 Einen Benutzer anlegen . . . 186
5.8.3.3 Abfragen . . . 187
5.8.4 phpMyAdmin . . . 188
5.8.5 Integration in Systemd . . . 193
5.8.6 MariaDB Log . . . 195

5.8.7 Quellen MariaDB 196
5.9 PHP einrichten 197
5.9.1 PHP im Dateisystem 197
5.9.2 PHP-Unterstützung für Apache2 197
5.9.3 PHP Konfiguration 198
5.9.4 PHP Module 199
5.9.5 Apache Log 202
5.9.6 Quellen PHP 202

6 Hardware 203
6.1 Grafiktreiber 203
6.1.1 Open Source Xorg-Treiber 203
6.1.2 Proprietäre Treiber 204
6.1.3 Videotreiber 2D 204
6.1.4 Videotreiber 3D 205
6.1.5 nVidia closed Source Treiber 205

7 Systemadministration 208
7.1 Terminal - Kommandozeile 210
7.1.1 Arbeit als root 211
7.1.2 Farbiges Terminal 213
7.1.3 Wenn das Terminal hängt 216
7.1.4 Hilfe im Terminal 216
7.1.5 Linux Konsolenbefehle 217
7.1.6 Skripte benutzen 217
7.2 Systemadministration allgemein 220
7.2.1 Bootoptionen Cheatcodes 220
7.2.2 systemd - Dienste verwalten 220
7.2.3 systemd.service 221
7.2.4 systemd - UNIT eingliedern 222
7.2.5 systemd-target ehemals Runlevel 222
7.2.6 Beenden eines Prozesses 224
7.2.7 Vergessenes Rootpasswort 225
7.2.8 Setzen neuer Passwörter 225
7.2.9 Schriftarten in siduction 226
7.2.10 Userkonfiguration 228

7.2.11 CUPS - das Drucksystem 230
7.2.12 Sound in siduction 231
7.3 Doas - Alternative zu Sudo 233
7.3.1 Doas konfigurieren 233
7.3.2 Doas und mehrere Benutzer 234
7.4 APT Paketverwaltung 237
7.4.1 apt und apt-get 237
7.4.2 sources.list - Liste der Quellen 238
7.4.3 apt update 240
7.4.4 Pakete installieren 241
7.4.5 Pakete entfernen 243
7.4.6 Hold oder Downgraden eines Pakets 244
7.4.7 Aktualisierung des Systems 246
7.4.8 Aktualisierbare Pakete 246
7.4.9 full-upgrade ausführen 249
7.4.10 Warum ausschließlich apt verwenden 249
7.4.11 Programmpakete suchen 250
7.4.11.1 Paketsuche im Terminal 251
7.4.11.2 Graphische Paketsuche 252
7.5 Lokaler APT-Mirror 255
7.5.1 Server installieren 255
7.5.2 Client Konfiguration 258
7.6 Kernel Upgrade 260
7.6.1 Kernel-Aktualisierung ohne Systemaktualisierung 260
7.6.2 Module 260
7.6.3 Entfernen alter Kernel 260
7.7 Systemd, der System- und Dienste-Manager 262
7.7.1 Konzeption des systemd 262
7.7.2 Unit Typen 263
7.7.3 Systemd im Dateisystem 264
7.7.4 Weitere Funktionen von systemd 264
7.7.5 Handhabung von Diensten 264
7.7.6 Quellen systemd 266
7.8 systemd unit-Datei 267
7.8.1 Ladepfad der Unit-Dateien 267
7.8.2 Aktivierung der Unit-Datei 268

7.8.3 Sektionen der Unit-Datei 269
7.8.3.1 Sektion Unit 269
7.8.3.2 Typ-spezifische Sektion 273
7.8.3.3 Sektion Install 273
7.8.4 Beispiel cupsd 275
7.8.5 Werkzeuge 279
7.8.6 Quellen systemd-unit-Datei 283
7.9 systemd-service 284
7.9.1 service-Unit anlegen 284
7.9.2 Sektion Service 284
7.9.3 Quellen systemd-service 288
7.10 systemd-mount 289
7.10.1 Inhalt der mount-Unit 290
7.10.2 Inhalt der automount-Unit 291
7.10.3 Beispiele 291
7.10.4 Quellen systemd-mount 296
7.11 systemd-target - Ziel-Unit 297
7.11.1 Besonderheiten 298
7.11.2 Quellen systemd-target 299
7.12 systemd-path 300
7.12.1 Benötigte Dateien 300
7.12.2 path-Unit Optionen 300
7.12.3 path-Unit anlegen 302
7.12.4 service-Unit für path 303
7.12.4.1 Zusätzliche service-Unit anlegen 304
7.12.5 path-Unit eingliedern 305
7.12.6 service-Unit manuell ausführen 306
7.12.7 Quellen systemd-path 307
7.13 systemd-timer 308
7.13.1 Benötigte Dateien 308
7.13.2 service-Unit für timer 308
7.13.3 timer-Unit anlegen 309
7.13.4 timer-Unit als cron Ersatz 311
7.13.5 Quellen systemd-timer 311
7.14 Systemjournal 312
7.14.1 journald 312

7.14.2 journald über das Netzwerk 313
7.14.3 journald.conf 313
7.14.4 journalctl 314
7.14.5 journalctl beherrschen 318
7.14.6 Quellen journald 323
7.15 Testpage for people with color vision deficiency 324
7.15.1 Warning 324
7.15.2 Code block 324
7.15.3 Inline code 324
7.15.4 Link 324

1 Willkommen

1.1 Das siduction™ GNU/Linux-Betriebssystem

Der Name **siduction** ist ein Wortspiel aus zwei Begriffen: Dem Wort **sid**, also dem Codenamen von Debian Unstable und **seduction**, im Sinne von verführen.

siduction ist ein Betriebssystem, das auf dem Linux-Kernel und dem GNU-Projekt basiert. Dazu kommen Anwendungsprogramme von Debian. siduction ist den Grundwerten des Debian Gesellschaftsvertrags und den daran anschließenden *"Debian Free Software Guidelines"* verpflichtet.
Siehe auch DFSG

1.1.1 Allgemeines

Für Schnellentschlossene geht es hier weiter zur Kurzanleitung

Das Handbuch des siduction Betriebssystems ist eine Referenz zum Kennenlernen des Systems wie auch zum Auffrischen der Kenntnisse über das System. Es vermittelt nicht nur Grundlagenwissen, sondern umfasst auch komplexe Themenkreise und unterstützt die Arbeit als Administrator von siduction-Systemen.

Es ist nach gleichartigen Themen unterteilt: Alles was zum Beispiel das Partitionieren betrifft, befindet sich im Kapitel "Installation/Partitionieren", und Themen, die WLAN betreffen befinden sich im Kapitel "Netzwerk".

Drucken von Handbuchseiten:
Linuxbefehle können mehr als 120 Zeichen lang sein. Für eine optimierte Darstellung am Bildschirm findet kein automatischer Zeilenumbruch statt. Unser Handbuch im PDF-Format, dass auf allen ISOs und nach der Installation im System verfügbar ist, verwendet für die langen Befehle den Zeilenumbruch. Es steht auch hier online zur Verfügung.
Zum Drucken von Handbuchseiten verwende bitte das PDF und drucke nur die benötigten Seiten.

Um Hilfe für ein spezifisches vorinstalliertes oder selbst installiertes Anwendungsprogramm (auch Paket genannt) zu erhalten, informiert man sich am besten in den FAQ, Online-Handbüchern oder Foren auf der Homepage bzw. im Hilfe-Menü der Anwendung.

Fast alle Anwendungsprogramme bieten Hilfestellung mittels einer zugehörigen *"Manual-Page"* (kurz Manpage). Sie wird im Terminal durch den Befehl `man <Paketname>` aufgerufen. Auch kann nachgesehen werden, ob sich eine Dokumentation in `/usr/share/doc/<paketname>` befindet.

1.1.2 Copyright Rechts- und Lizenzhinweise

Alle Rechte © 2006-2021 des siduction-manual sind lizenziert unter der GNU Free Documentation License. Eine informelle Übersetzung dieser Lizenz ins Deutsche befindet sich hier.
Dies gestattet das Dokument nach den Bestimmungen der GNU Free Document License Version 1.3 oder neuer (wie veröffentlicht bei der Free Software Foundation) zu kopieren, verbreiten und/oder zu ändern; ohne unveränderliche Sektionen und ohne Umschlagtexte (Vorderseitentexte, Rückseitentexte).

Die Rechte von geschützten Marken bzw. Urheberrechte liegen bei den jeweiligen Inhabern, unabhängig davon, ob dies vermerkt ist oder nicht.

Irrtum vorbehalten (E&OE)

1.1.3 Haftungsausschluss

Dies ist experimentelle Software. Benutzung geschieht auf eigenes Risiko. Das siduction-Projekt, seine Entwickler und Teammitglieder können unter keinen Umständen haftbar gemacht werden für Schäden an Hard- oder Software, Datenverlust oder anderen, direkten oder indirekten Schäden, entstanden durch die Benutzung dieser Software.

Solltest Du mit diesen Bedingungen nicht einverstanden sein, so ist es Dir nicht gestattet, diese Software weiter zu benutzen oder zu verteilen.

Zuletzt bearbeitet: 2021-11-28

1.2 Das siduction-Team

Alphabetisch nach Familiennamen bzw. Pseudonym sortierte Liste der Maintainer und Autoren, die sich für die Entwicklung, den Erhalt und die Unterstützung von **siduction** einsetzen und einsetzten.

Über dieses Kontaktformular erreichst du das siduction-Team.

1.2.1 Credit für siduction 2021.2.0

1.2.1.1 Core Team

- Axel Beu (ab) 2021 †
- Ferdinand Thommes (devil)
- Hendrik Lehmbruch (hendrikL)
- Torsten Wohlfarth (towo)
- Vinzenz Vietzke (vinzv)
- Alf Gaida (agaida)

1.2.1.2 Art Team

- hendrikL

We **need** contributors for siduction release art!

1.2.1.3 Code Ideen Unterstützung Handbuch

- der_bud
- Markus Meyer (coruja)
- A.Konrad (akli) (for his work on getting the manual back in shape)
- Stefan Tell (cryptosteve)

1.2.1.4 Credit für das original manual Team

- Trevor Walkley (bluewater)
- Jose Tadeu Barros (ceti)

- Alpha Mohamed Diakite (alphad)
- Stefan R. Eissens (eislon)
- Roland Engert (RoEn)
- Alessio Giustini (alessiog75)
- Markus Huber (hubi)
- Luis_P
- Janusz Martyniak (wiarus_old)
- Philippe Masson (LjanA)
- Mutsumu Nomura (muchan)
- Rasmus Güllich Pørksen (ragupo)
- Dawid Staropietka (DaVidoSS)
- Bruno Torremans (btorrem)
- Robert Ulatowski (quidam77)
- Dorin Vatavu (dorin)
- Bram Verdoodt (Bram0s)
- Petr Vorel (pumrel)
- zenren

Wir möchten allen, die zu siduction beigetragen haben und weiter beitragen genauso danken, wie den ursprünglichen Erstellern und Übersetzern des bluewater-manual

Zuletzt bearbeitet: 2021-07-27

1.3 siduction Hilfe

Schnelle Hilfe kann einem viele Tränen ersparen und bietet die Möglichkeit, das weiter zu machen, was wirklich wichtig ist im Leben. Dieses Thema ist nach Bereichen gegliedert, wo die Distribution siduction Hilfe anbietet:

1.3.1 Das siduction-Forum

Das siduction-Forum bietet die Möglichkeit Fragen zu stellen und Antworten auf diese zu erhalten. Bevor ein neuer Beitrag erstellt wird, sollte die Forensuche benutzt werden, da die Wahrscheinlichkeit groß ist, dass diese oder eine ähnliche Frage schon einmal gestellt wurde. Das Forum ist auf Deutsch und Englisch verfügbar.

1.3.2 Das siduction-Wiki

Das siduction-Wiki ist von allen siduction-Nutzern frei nutz- und veränderbar. Wir hoffen so, dass die siduction-Dokumentation im Laufe der Zeit mit dem Projekt wachsen wird.

Wir hoffen auf Beiträge von Linuxnutzern aller Erfahrungsebenen, da dieses Wiki beabsichtigt, Nutzern jeden Kenntnisstandes zu helfen. Die wenigen Minuten, die dem Wiki und Projekt “geopfert” werden, können anderen Nutzern (und vielleicht einem selbst) Stunden des Suchens nach Problemlösungen ersparen. Link zum siduction-Wiki .

1.3.3 IRC - interaktiver Livesupport

Der IRC soll nie als “root” betreten werden, sondern nur als normaler Nutzer. Bei Unklarheiten bitte dies sofort im IRC-Channel bekannt geben, damit Hilfe gegeben werden kann.

Verhaltensregeln im IRC

- Ein freundlicher Umgangston ist obligatorisch, denn wir leisten den Support alle ehrenamtlich.
- Hilfreich ist es, eine nach bestem Wissen genaue Anfrage zu stellen und nach Möglichkeit zuvor im siduction-Wiki nach Lösungen zu suchen.

- Bitte niemals gleichzeitig im IRC und Forum eine Anfrage stellen. Bestenfalls reiben wir uns verwundert die Augen.

siduction erreichen

- Klicke einfach auf das **“IRC Chat #siduction”-Symbol** auf dem Desktop oder verwende den kmenu-Eintrag von koversation.
  Wenn du einen anderen Chat-Client bevorzugst, musst du diese Serverdaten eingeben:

```
irc.oftc.net
port 6667
```

- Mit diesem Link kannst Du den IRC sofort in Deinem Browser aufrufen : gib dazu einen frei gewählten Nicknamen ein und betritt den Channel #siduction-de.

1.3.4 Nützliche Helfer im Textmodus

Normalerweise verwendet man den Textmodus Runlevel 3 (`init 3` bzw. `journalctl isolate multi-user.target`), wenn man ein dist-upgrade durchführen möchte, oder gezwungenermaßen, wenn das System einen schwerwiegenden Fehler aufweist.

gpm

ist ein hilfreiches Programm im Textmodus. Dieses ermöglicht, die Maus zum Kopieren und Einfügen im Terminal zu benutzen.

gpm ist in siduction vorkonfiguriert. Falls dem nicht so ist:

```
$ gpm -t imps2 -m /dev/input/mice
```

Danach sollte man prüfen ob der Service aktiv ist:

```
$ systemctl status gpm.service
```

Bei Erfolg findet sich in der Ausgabe auch eine Zeile ähnlich der folgenden.

```
  Active: active (running) since Thu 2020-04-09 12:17:14 CEST⇙
     ; 5min ago
```

Nun sollte man seine Maus im Textmodus (tty) nutzen können.

Dateimanager und Textbearbeitung

Der *"Midnight Commander"* ist ein einfach zu bedienender Dateimanager im Text-Modus (tty) und Texteditor. Er wird mit siduction ausgeliefert.
Abgesehen von den normalen Tastatureingaben kann aufgrund von gpm auch die Maus benutzt werden.
Mit dem Befehl `mc` wird der Midnight Commander aufgerufen und mit `mcedit` kann eine vorhandene Datei bearbeitet bzw. eine neue Datei erstellt werden.

So öffnet man eine vorhandene Datei (zuerst wird eine Sicherungskopie angelegt):

```
$ cp /etc/apt/sources.list.d/debian.list /etc/apt/sources.↙
   list.d/debian.list_$(date +%F)

  anschließend

$ mcedit /etc/apt/sources.list.d/debian.list
```

Nun kann die Datei bearbeitet und gespeichert werden. Die Änderungen werden sofort wirksam.
Weitere Informationen auf der Manpage:

```
$ man mc
```

1.3.5 siduction IRC-Support im Textmodus

Verhaltensregel im IRC

Der IRC soll nie als "root" betreten werden, sondern nur als normaler Nutzer.
Bei Unklarheiten bitte dies sofort im IRC-Channel bekannt geben, damit Hilfe gegeben werden kann.

IRC im Textmodus

Das Programm `irssi` stellt einen IRC-Client im Textmodus oder der Konsole bereit und ist in siduction aktiviert.
Mit der Tastenkombination **ALT**+**F2** oder **F3** usw. kann man von einem Terminal/TTY in ein anderes wechseln und sich dort mit seinem Useraccount anmelden:

```
$ siductionbox login: <username> <password> (nicht als root)
```

danach gibt man

```
$ siduction-irc
```

ein, um irssi zu starten.

Anleitung, falls ein anderer Client (im Beispiel weechat) gewünscht ist:
Zuerst stellt man sicher, dass WeeChat installiert ist, indem man im Menü den Eintrag von weechat sucht. Falls dieser nicht vorhanden sein sollte:

```
# apt update
# apt install weechat-curses

  und anschließend das Programm starten

$ weechat-curses
```

Jetzt kann man sich mit irc.oftc.net auf Port 6667 verbinden. Nach erfolgter Verbindung wird das Pseudonym (der “Nickname”) geändert:

```
/nick 'Dein_neuer_nick'
```

Den siduction-Channel betritt man mit folgender Eingabe:

```
/join #siduction-de
```

Falls man wünscht, den Server zu wechseln, gibt man einen Befehl mit folgender Syntax ein:

```
/server server.name
```

In der unteren Menüzeile sieht man Zahlen, falls die Channel aktiv sind, und um sich mit einem Channel zu verbinden, verwendet man ALT-1, ALT-2, ALT-3, ALT-4 usw.

Einen Channel verlässt man mit

```
/exit
```

Falls gleichzeitig ein dist-upgrade durchgeführt wird, kann man folgendermaßen das Terminal wechseln, um den Fortschritt des Upgrades zu verfolgen:

Tastenkombination **ALT**+**F3**
und zum IRC kommt man zurück mit der
Tastenkombination **ALT**+**F2**

Die folgenden Link bieten weitere Informationen.
Dokumentationsseite von irssi (Englisch)
Dokumentationsseite von WeeChat (Deutsch)

1.3.6 Surfen im Internet im Textmodus

Der Kommandozeilenbrowser `w3m` ermöglicht das Surfen im Internet in einem Terminal bzw. einer Konsole oder im Textmodus

Falls w3m oder elinks nicht installiert sind, geht man so vor:

```
# apt update
# apt install w3m
# apt install elinks
```

Nun kann man den Kommandozeilenbrowser w3m benutzen. Dazu ist es sinnvoll in ein anderes Terminal zu wechseln und sich mit seinem Useraccount anzumelden:

Tastenkombination **ALT**+**F2**

```
$ siductionbox login: <username> <password> (nicht root!)
```

Der Programmaufruf lautet `w3m URL` oder `w3m ?`.
Beispiel: https://siduction.org ruft man so auf (https:// wird weggelassen):

```
$ w3m siduction.org
```

Eine neue URL wird mit Hilfe der Tastenkombination Shift+U aufgerufen:

SHIFT+**U**

Danach sieht man eine Zeile wie `Goto URL: https://siduction.org`. Mit der Rücktaste löscht man die zuletzt gewählte URL und gibt die gewünschte ein.
Beendet wird w3m mit:

SHIFT+**Q**

Mehr Informationen gibt es auf der Dokumentationsseite von w3m (Englisch)

Es ist ratsam, sich vor einem Notfall mit *“elinks/w3m”*, *“irssi/weechat”*, *“midnight commander”* vertraut zu machen. Drucke diese Datei aus, um im Notfall die Informationen griffbereit zu haben.

1.3.7 inxi

Inxi ist ein System-Informations-Skript, welches unabhängig von einzelnen IRC-Clients funktioniert. Dieses Skript gibt verschiedene Informationen über die benutzte Hard- und Software aus, sodass andere Nutzer in #siduction bei der Fehlerdiagnose besser helfen können. Oder in einer Konsole ausgeführt, kann man selbst Informationen über das eigene System erhalten.

Um inxi in Konversation zu nutzen, gibt man in die Chatbox dies ein:

```
/cmd inxi -v2
```

Um inxi in weechat zu nutzen, gibt man in die Chatbox dies ein:

```
/shell -o inxi -v2
```

Vorausgesetzt, dass man die Erweiterung “shell” installiert hat.
Siehe dazu: https://www.weechat.org/scripts/
Um inxi in anderen Klienten zu nutzen, gibt man in die Chatbox dies ein:

```
/exec -o inxi -v2
```
oder
```
/inxi -v2
```

In einer Konsole wird folgender Befehl eingegeben:

```
$ inxi -v2
```

Hilfe zu inxi

```
$ inxi --help
```

1.3.8 Nützliche Links

Debian Referenzkarte - zum Ausdruck auf ein Einzelblatt
HOWTOs von der Debian-Seite (ist automatisch auf Deutsch, wenn Browser loka-

lisiert ist)

Debian-Referenz: Grundlagen und Systemadministration Dokumente verfügbar als HTML, Text, PDF und PS

Common Unix Printing System CUPS (EN) . In KDE bietet das KDE-Hilfezentrum Informationen zu CUPS.

LibreOffice Im Menü “Hilfe” finden sich viele Angebote.

Zuletzt bearbeitet: 2022-01-13

2 Quickstart

2.1 siduction Kurzanleitung

siduction strebt danach, zu 100% mit Debian Sid kompatibel zu sein. Trotzdem kann siduction gegebenenfalls Pakete anbieten, welche temporär fehlerhafte Debian-Pakete ersetzen. Das Apt-Repository von siduction enthält siduction spezifische Pakete wie den siduction-Kernel, Skripte, Pakete, die wir gern nach Debian pushen würden, Hilfsprogramme und Dokumentationen.

2.1.1 Essenzielle Kapitel

Einige Kapitel des Handbuchs stellen für Nutzer, die neu bei Linux bzw. neu bei siduction sind, essenzielle Lektüre dar. Neben dieser Kurzeinführung sind das:

- Terminal/Konsole - Beschreibt, wie ein Terminal und der su-Befehl zu nutzen sind.
- Partitionieren der Festplatte - Beschreibt, wie eine Festplatte partitioniert werden kann.
- siduction ISO herunterladen - Beschreibt den Download und die Prüfung einer siduction ISO-Datei.
- ISO auf USB-Stick - Speicherkarte - Beschreibt wie ein siduction Live-Medium erstellt wird.
- Installation auf einer Festplatte - Beschreibt, wie siduction auf einer Festplatte installiert wird.
- Nicht freie Treiber, Firmware und Quellen - Beschreibt, wie Softwarequellen adaptiert und nicht freie Firmwares installiert werden können.
- Internetverbindung - Beschreibt, wie man sich mit dem Internet verbinden kann.
- Paketmanager und Systemaktualisierung - Beschreibt, wie neue Software installiert und das System aktualisiert werden kann.

2.1.2 Zur Stabilität von Debian Sid

"*Sid*" ist der Name des Unstable-Repositories von Debian. Debian Sid wird regelmäßig mit neuen Softwarepaketen beschickt, wodurch diese Debian-Distribution sehr zeitnah die neuesten Versionen der jeweiligen Programme enthält. Dies bedeutet aber auch, dass zwischen einer Veröffentlichung im Upstream (von den Softwareentwicklern) und der Verteilung in Debian Sid weniger Zeit ist, um die Pakete zu testen.

2.1.3 Der siduction-Kernel

Der Linux-Kernel von siduction ist optimiert, um folgende Ziele zu erreichen: Problembehebung, erweiterte und aktualisierte Funktionen, Leistungsoptimierung, höhere Stabilität. Basis ist immer der aktuelle Kernel von http://www.kernel.org/ .

2.1.4 Die Verwaltung von Softwarepaketen

siduction richtet sich nach den Debian-Regeln bezüglich der Paketestruktur und verwendet `apt` und `dpkg` für das Management der Softwarepakete. Die Repositorien von Debian und siduction befinden sich in `/etc/sources.list.d/*`

Debian Sid enthält mehr als 20.000 Programmpakete, womit die Chancen, ein für eine Aufgabe geeignetes Programm zu finden, sehr gut stehen. Wie man Programmpakete sucht, ist hier beschrieben:
Programmsuche mit apt-cache bzw. apt
oder mit
GUI-Paketsuche mit packagesearch .

Ein Programmpaket wird mit diesem Befehl installiert:

```
apt install <Paketname>
```

Siehe auch: Neue Pakete installieren .

Die Repositorien von Debian Sid werden in der Regel viermal am Tag mit aktualisierten bzw. neuen Softwarepaketen beschickt. Zur schnellen Verwaltung der Pakete wird eine lokale Datenbank verwendet. Der Befehl

```
apt update
```

ist vor jeder Neuinstallation eines Softwarepakets notwendig, um die lokale Datenbank mit dem Softwareangebot der Repositorien zu synchronisieren.

Die Nutzung anderer auf Debian basierender Repositorien, Quellen und RPMs
Installationen aus Quellcode sind nicht unterstützt. Empfohlen ist eine Kompilierung als **user** (nicht als root) und die Platzierung der Anwendung im Home-Verzeichnis, ohne dass sie ins System installiert wird. Die Verwendung von `checkinstall` zum Erzeugen von DEB-Paketen sollte auf die rein private Nutzung beschränkt bleiben. Konvertierungsprogramme für RPM-Pakete wie `alien` sind nicht empfohlen.

Andere bekannte (und weniger bekannte) Distributionen, die auf Debian basieren, erstellen neue, von Debian verschieden strukturierte Pakete und verwenden oft andere Verzeichnisse, in denen bei der Installation Programme, Skripte und Dateien abgelegt werden, als Debian. Dies kann zu instabilen Systemen führen. Manche Pakete lassen sich wegen nicht auflösbarer Abhängigkeiten, unterschiedlicher Benennungskonventionen oder unterschiedlicher Versionierung überhaupt nicht installieren. Eine unterschiedliche Version von `glibc` zum Beispiel kann dazu führen, dass kein Programm lauffähig ist.

Aus diesem Grund sollen die Repositorien von Debian benutzt werden, um die benötigten Softwarepakete zu installieren. Andere Softwarequellen können nur schwer oder gar nicht von siduction unterstützt werden. Darunter fallen auch Pakete und PPAs von Ubuntu.

2.1.5 Aktualisierung des Systems - upgrade

Ein upgrade ist nur bei beendetem Grafikserver X durchzuführen. Um den Grafikserver zu beenden, gibt man als **root** den Befehl

```
init 3
```

in eine Konsole ein. Danach sind Systemaktualisierungen sicher durchführbar. Zuerst die lokale Paketdatenbank auffrischen mit

```
apt update
```

dann mit einer der beiden Varianten das System aktualisieren.

```
apt upgrade
apt full-upgrade
```

Anschließend startet man mit folgendem Befehl wieder die graphische Oberfläche:

```
init 5
```

"apt full-upgrade" ist das empfohlene Verfahren, um eine siduction-Installation auf den neuesten Stand zu bringen. Ausführlicher wird das hier beschrieben: Aktualisierung eines installierten Systems - full-upgrade.

2.1.6 Konfiguration von Netzwerken

Der in allen graphischen Oberflächen von siduction integriert `Networkmanager` bietet eine schnelle Konfiguration von Netzwerkkarten (Ethernet und drahtlos). Er ist größtenteils selbsterklärend. Im Terminal bietet das Skript `nmcli` Zugang zur Funktionalität des Netwokmanagers. Drahtlose Netzwerke werden von dem Skript gescannt, man kann die Verschlüsselungsmethoden WEP und WPA wählen und die Backends `wireless-tools` bzw. `wpasupplicant` zur Konfiguration drahtloser Netzwerke verwenden. Die Ethernet-Konfiguration erfolgt bei Verwendung eines DHCP-Servers am Router (dynamische Zuweisung einer IP-Adresse) automatisch, aber auch die Möglichkeit eines manuellen Setups (von Netmasks bis Nameserver) ist mit diesem Skript gegeben.

Der Startbefehl in der Konsole ist **`nmcli`** oder **`nmtui`** . Falls das Skript nicht vorhanden ist, installiert man es mit:

```
apt install network-manager
```

Mehr Informationen unter Netzwerk - nmcli

Intels iNet wireless daemon (iwd) schickt sich an, den WPA-Supplicant in den wohlverdienten Ruhestand zu verabschieden. Nur ein Zehntel so groß und viel schneller, ist iwd der Nachfolger. Wer schon jetzt zum iwd wechseln möchte, informiert sich bitte auf unserer Handbuchseite IWD statt wpa_supplicant über die Vorgehensweise.

2.1.7 Runlevels - Ziel-Unit

Standardmäßig bootet siduction in die graphische Oberfläche (außer NoX).
Die Konfiguration der Runlevel ist im Kapitel siduction-Runlevels - Ziel-Unit beschrieben.

2.1.8 Weitere Desktopumgebungen

Plasma, Xfce, LXQt und Xorg werden von siduction ausgeliefert.

2.1.9 Hilfe im IRC und im Forum

Hilfe gibt es jederzeit im IRC bzw. im Forum von siduction.

- Mehr dazu im Kapitel Wo es Hilfe gibt .
- Mit diesem Link kannst Du den IRC sofort in Deinem Browser aufrufen : gib dazu einen frei gewählten Nicknamen ein und betritt den Channel #siduction-de.

Zuletzt bearbeitet: 2022-16-04

3 ISO-Abbilder

Dieser Abschnitt beinhaltet Informationen und Hinweise zum/zur

- Inhalt der Live-ISO,
  den verfügbaren Varianten, den Systemanforderungen, zu Anwendungen und Hilfsprogrammen und zum Haftungsausschluss.
- Verwendung der Live-DVD,
  den eingerichteten Usern und deren Passwörtern, der Arbeit mit root-Rechten und der Software-Installation während der Live-Sitzung.
- Den Bootoptionen (Cheatcodes)
  in tabellarischer Form sowohl für die Live-ISO als auch für installierte Systeme.
- Download und Integritätsprüfung der ISOs
  mit Angaben zu den Spiegelservern und den darauf befindlichen Dateien und der Integritätsprüfung des Download.
- Schreiben der ISOs auf USB-Stick, SD- oder SDHC-Karte
  Methoden eine siduction ISO-Abbilddatei als Live-Medium auf einen USB-Stick, eine SD-Karte oder eine SDHC-Karte zu schreiben.
- Brennen der ISOs auf DVD mit GUI
  Programme der graphischen Oberfläche unter den Betriebssystemen Linux und Windows™ verwenden.
- Brennen der ISOs auf DVD ohne GUI
  unter Linux mit einer ganzen Reihe direkt verwendbarer Terminal-Befehle auch zur Ermittlung der verfügbaren Geräte.

Zuletzt bearbeitet: 2022-03-11

3.1 Inhalt der Live-ISO

3.1.1 Hinweis zur Software auf dem Live-ISO

siduction stellt auf der Live-ISO DFSG-freie Software zur Verfügung als auch nicht freie Firmware. Zur Deinstallation proprietärer Software benutzt man den Befehl `apt purge $(vrms -s)` oder unser Script `remove-nonfree` nach der Installation.

Das ISO basiert ausschließlich auf zum Veröffentlichungszeitpunkt jeweils aktuellem Debian Sid, bereichert und stabilisiert durch eigene Pakete und Skripte aus den siduction-Repositories. Als Kernel wird der jeweils aktuelle Vanilla Mainline Kernel verwendet und mit Patches versehen. ACPI und DMA sind aktiviert.

Eine komplette Manifest-Datei mit der Auflistung aller installierten Programme für jede einzelne Veröffentlichungs-Variante von siduction findet man auf jedem Download-Spiegelserver.

3.1.2 Varianten der ISO

siduction bietet fünf aktuelle Images und verschiedenen Desktop-Umgebungen (ein Image ohne) in 64-Bit als Live-ISO zum Einstieg in Debian Sid. Üblicherweise dauert eine Installation zwischen 1 und 10 Minuten, je nach Hardware.
Die Varianten sind:

1. **KDE 64 Bit** , live-ISO mit etwa 2,8 GByte:
   - Qt basierter Plasma Desktop und KDE-Frameworks. Mit einer repräsentativen Auswahl der KDE Applications.
   - Die Installation zusätzlicher Anwendungen ist ohne Probleme via apt möglich.
2. **XFCE 64 Bit** , live-ISO mit etwa 2,3 GByte:
   - umfasst eine GTK basierte Desktop-Umgebung mit allen Features (keine Minimalversion!) und alle Anwendungen um sofort produktiv tätig sein zu können.
   - Der Ressourcenaufwand ist geringer als mit KDE.
   - Die Installation zusätzlicher Anwendungen ist ohne Probleme via apt möglich.

3. **LXQt mit 64 Bit** , live-ISO mit etwa 2,2 GByte:
    - umfasst eine Desktopumgebung mit einer Auswahl an Qt-Applikationen.
    - Der Fußabdruck ist etwas schmaler als bei XFCE
    - Die Installation zusätzlicher Anwendungen ist ohne Probleme via apt möglich.
4. **Xorg mit 64 Bit** , live ISO mit etwa 1,8 GByte:
    - Ein ISO-Image mit einem Xorg-Stack und dem spartanischen Fenstermanager Fluxbox.
    - Für Anwender, die sich ihr System nach eigenen Vorstellungen aufbauen wollen
5. **NoX mit 64 Bit** , live-ISO mit etwa 800 MByte:
    - Wie der Name andeutet: kein vorinstallierter Xorg-Stack

32 Bit ISOs bieten wir standardmäßig nicht mehr an.
Wenn ein 32 Bit IOS gewünscht ist, wird ein solches auf Anfrage im IRC gerne erstellt. Testen können wir ein solches ISO leider nicht.

3.1.3 Minimale Systemanforderungen

für: KDE-Plasma, XFCE, LXQt, Xorg und NoX

Prozessoranforderungen: 64 Bit CPU

```
AMD64
Intel Core2
Intel Atom 330
jede x86-64/ EM64T fähige CPU oder neuer
neuere 64 Bit fähige AMD Sempron und Intel Pentium 4 CPUs
(achten Sie auf das "lm"-Flag in /proc/cpuinfo oder nutzt ↙
   inxi -v3).
```

Speicheranforderungen

```
KDE-Plasma: ≥ 4 GByte RAM
XFCE: ≥       4 GByte RAM
LXQT: ≥       2 GByte RAM
```

```
Xorg: ≥       512 MByte RAM
NoX: ≥        256 MByte RAM≥

 5 GByte Festplattenspeicher für NOX≥
10 GByte Festplattenspeicher für alle Anderen
```

Sonstiges

```
VGA Grafikkarte mit mindestens 640x480 Pixel Auflösung.
optisches Laufwerk oder USB Medien.
```

3.1.4 Anwendungen und Hilfsprogramme

Als Internetbrowser werden (je nach Variante) Firefox, oder Chromium mitgeliefert.

Als Bürosoftware ist Libreoffice vorinstalliert. Als Dateimanager stehen unter anderem Dolphin, Thunar und PCManFM zur Verfügung.

Zur Netzwerk- und Internetkonfiguration steht Connman, Network-Manager oder IWD zur Verfügung.

Xorg und nox werden mit IWD als ausgeliefert, dieser kann via nmtui/nmcli oder iwctl konfiguriert werden.

Zur Partitionierung von Festplatten werden cfdisk, gdisk und cgdisk und GParted mitgeliefert. Gparted bietet auch die Möglichkeit, die Größe von NTFS-Partitionen zu ändern.

Tools zur Systemanalyse wie Memtest86+ (ein Tool zur umfassenden Speicheranalyse) werden ebenso mitgeliefert.

Jede ISO-Variante enthält eine umfangreiche Auswahl an Anwendungen für die Befehlszeile. Eine komplette Manifest-Datei mit den installierten Programmen für jede einzelne Veröffentlichungsvariante von siduction findet man auf jedem Download-Spiegelserver.

3.1.5 Haftungsausschluss_Disclaimer

siduction ist experimentelle Software. Benutzung auf eigene Gefahr. Das siduction-Projekt, seine Entwickler und Team-Mitglieder können unter keinen Umständen

wegen Beschädigung von Hardware oder Software, verlorener Daten oder anderer direkter oder indirekter Schäden des Nutzers durch Nutzung dieser Software zur Rechenschaft gezogen werden. Wer diesen Bedingungen nicht zustimmt, darf diese Software weder verwenden noch verteilen.

Zuletzt bearbeitet: 2021-11-28

3.2 Live-DVD verwenden

3.2.1 Eingerichtete User auf dem Live-Medium

Auf dem Live-Medium sind die User **siducer** und **root** (der Systemadministrator) eingerichtet.

Für den User **siducer** ist das Passwort **live** gesetzt.
Für den User **root** (Systemadministrator) ist kein Passwort gesetzt.

Die Live-Session wird nach geraumer Zeit ohne Eingaben gesperrt. Zum Entsperren bitte den User **siducer** mit dem Passwort **live** eingeben.

3.2.2 Mit root-Rechten auf der Live-DVD

Wir beschreiben nachfolgend mehrere Möglichkeiten, ein Programm mit root-Rechten auszuführen.

Achtung:
Wann immer man mit root-Rechten arbeitet, sollte man genau wissen, was man macht. Für das Surfen im Internet und ähnliche Aktionen sind keine root-Rechte nötig.

1. Am einfachsten man öffnet ein Terminal und verschafft sich mit der Eingabe **su** root-Rechte.
   Um jetzt ein Programm, das mit graphischer Oberfläche arbeitet zu starten, einfach den Programmnamen eingeben.

   ```
   root@siduction:~# geparted &
   ```

   Jetzt wird Gparted mit root-Rechten ausgeführt. Das *"&"* am Ende des Befehls bringt den Prozess in den Hintergrund und das Terminal bleibt weiter benutzbar.

2. Ein Befehlseingabefenster öffnen:
   Die Tastenkombination **Alt**+**F2** benutzen um eine Programmstartzeile zu erhalten und darin den Befehl

   ```
   sudo <Anwendung>
   ```

eingeben.
Es öffnet sich ein Terminalfenster, in dem das root-Passwort abgefragt wird. Nun einfach die **Enter**-Taste betätigen, es sei denn, es wurde wie weiter unten beschrieben ein temporäres root-Passwort gesetzt, das einzugeben ist.

3. In ein Terminal ohne root-Rechte den Befehl

```
sudo <Anwendung> &
```

eingeben.

Bitte beachten:
sudo ist auf Festplatteninstallationen nicht vorkonfiguriert. Wir empfehlen, den echten root-Account direkt zu nutzen.
Siehe Arbeit als root, warum sudo nicht konfiguriert ist

3.2.3 Ein neues Passwort setzen

Noch einmal zur Erinnerung: Der Standardbenutzer in der Livesession ist **siducer** mit dem Passwort **live**. Will man das Passwort ändern, so öffnet man ein Terminal und gibt die folgenden Befehle ein:

```
siducer@siduction:~$ passwd
Geben Sie ein neues Passwort ein:
Geben Sie das neue Passwort erneut ein:
passwd: Passwort erfolgreich geändert
siducer@siduction:~$
```

Dieses neue Passwort für **siducer** kann den Rest der Live-Sitzung verwendet werden.

Mit der gleichen Prozedur kann man in jedem Terminal auch für **root** ein Passwort vergeben, allerdings muss man vorher per `su` **root** werden. Im Anschluss ist eine Anmeldung auf einer virtuellen Konsole als **root** möglich.

3.2.4 Software-Installation bei Live-Sitzung

Die Befehlsfolge für die Installation von Software während einer Live-Sitzung gleicht der bei einer Festplatteninstallation. Voraussetzung ist ein root-Terminal,

```
apt update
apt install <das-gewünschtes-paket>
```

oder ein vorangestelltes `sudo` vor die Befehle.

```
sudo apt update
sudo apt install <das-gewünschtes-paket>
```

Allerdings gilt: Wenn Du die Live-DVD herunterfährst, werden keine Änderungen behalten.

Zuletzt bearbeitet: 2022-01-13

3.3 Bootoptionen Cheatcodes

Info

Diese Handbuchseite enthält Tabellen zu den Bootoptionen für

1. siduction spezifische Parameter (nur Live-DVD)
2. Bootoptionen für den Grafikserver X
3. Allgemeine Parameter des Linux-Kernels
4. Werte für den allgemeinen Parameter **vga**

Sofern in dem “Werte”-Feld der Tabellen Werte aufgelistet werden, müssen diese an die betreffende Bootoption mit einem = Zeichen angehängt werden. Wenn zum Beispiel “1280x1024” der gewünschte Wert für die Bootoption “screen” wäre, dann wird `screen=1280x1024` in die Grub-Befehlszeile eingegeben, für die Sprachauswahl (hier “Deutsch”) `lang=de`. Die Grub-Befehlszeile lässt sich editieren, indem man, sobald das Grub-Menue erscheint, die Taste **e** drückt. Danach befindet man sich im Editiermodus. Jetzt kann man mit den Pfeiltasten zur Kernelzeile navigieren und am Ende den oder die gewünschten Cheatcode einfügen. Als Trennzeichen dient das Leerzeichen. Der Bootvorgang wird mit der Tastenkombination **Strg**+**X** oder **F10** fortgesetzt.

Ausführliche Referenzliste für Kernel-Bootcodes von kernel.org (Englisch, PDF)

3.3.1 siduction spezifische Parameter

Diese Bootoptionen gelten nur für die Live-DVD.

Bootoption	Wert	Beschreibung
blacklist	Name des Moduls	temporäre Deaktivierung von Modulen, bevor udev aktiv wird
desktop	kde, gnome, fluxbox	Desktopumgebung auswählen
fromiso		bitte lies “Booten ‘fromiso’ ”
hostname	myhostname	ändert den Netzwerknamen (hostname) des Live-CD-Systems

Bootoption	Wert	Beschreibung
lang	be, bg, cz, da, de, de_CH, el, en, en_AU, en_GB, en_IE, es, fr, fr_BE, ga, hr, hu, it, ja, nl, nl_BE, pl, pt (pt_BR), pt_PT, ro, ru, zh	setzt die Spracheinstellung, die Grundeinstellungen der Lokalisation (locales), das Tastaturlayout (in der Konsole wie in X), die Zeitzone und den Spiegelserver von Debian. Mit der Langform **lang=ll_cc** oder **lang=ll-cc** bedeutet **ll** die Sprachauswahl und **cc** Tastaturlayout, Spiegelserver und Zeitzonenwahl (z.B. “lang=fr-be”). Die Grundeinstellung für Englisch ist en_US mit UTC als Zeitzone und für Deutsch, de mit Europe/Berlin als die Zeitzone. Beispiel für eine selbstgewählte Einstellung: “lang=pt_PT tz=Pacific/Auckland”
md5sum		testet die Prüfsumme der CD/DVD (zur Kontrolle, ob CD/DVD in Ordnung sind)
noaptlang		Verhindert die Installation von Lokalisierungspaketen der gewählten Sprache
nocpufreq		aktiviert kein Speedstep/Powernow
nodhcp		kein DHCP (DHCP versucht automatisch Ethernetverbindungen aufzubauen)
noeject		entfernt CD/DVD nicht aus dem Laufwerk
nofstab		Verhindert das Schreiben einer neuen fstab
nointro		überspringt die Ausgabe der index.html beim Start der Live-DVD/CD

Bootoption	Wert	Beschreibung
nomodeset	radeon.modeset=0	ermöglicht zusammen mit **xmodule=vesa** ein sauberes Booten nach X bei Radeonkarten im Live-Mode
nonetwork		verhindert die automatische Konfiguration von Netzwerkschnittstellen beim Booten
noswap		Keine Aktivierung der Swap-Partition
persist		bitte lies “fromiso und persist”
smouse		sucht mittels hwinfo nach seriellen Mauseingabegeräten
tz	tz=Europe/Dublin	setzt die Zeitzone. Falls die Bios- bzw. Hardwareuhr auf UTC eingestellt ist, wird **utc=yes** angegeben. Eine Liste aller unterstützter Zeitzonen kann eingesehen werden, wenn per copy & paste: **file:///usr/share/zoneinfo/** in den Browser eingegeben wird .
toram		kopiert die DVD/CD ins RAM und startet aus der RAM-Kopie

3.3.2 Bootoptionen für den Grafikserver X

Es sollte zusätzlich auch entweder die Bootoption xandr oder xmodule verwendet werden, wenn man Bootoptionen für den Grafikserver X für die Grafikkarten Radeon, Intel oder MGA einsetzt.

Bootoption	Wert	Beschreibung
dpi	auto *oder* DPI-Zahl	setzt die gewünschten Pixel pro Zoll für den Monitor. Die DPI für den Monitor erhält man, wenn man die Pixelanzahl der Monitorbreite durch den Zollwert der Diagonale dividiert und mit folgenden Werten multipliziert: 1,25 für einen 4:3-Bildschirm, 1,18 für einen 16:10-Bildschirm oder 1,147 für einen 16:9-Bildschirm. Für einen 24“-Bildschirm mit der Auflösung 1920x1080 ergibt das mittels 1,147x1920/24 dpi=92 oder für einen 15”-Bildschirm mit der Auflösung 1600x1200 ergibt das mittels 1.25x1600/15 dpi=133.
hsync	80	setzt die horizontale Frequenz des Monitors (in Kilohertz)
noml		verhindert, dass die X.org-Konfiguration eine Liste von Modelines enthält, und bewirkt dadurch, dass der korrekte Mode automatisch erkannt wird
noxrandr		verhindert die Verwendung der Erweiterungen von RandR 1.2 durch die neuen X.org-Treiber und nutzt die alten Techniken zur Abfrage der Monitoreigenschaften
screen	1280x1024	stellt benutzerdefinierte Auflösung für X ein (1280x1024 oder andere Bildschirmauflösungen)
vsync	60	setzt die vertikale Frequenz des Monitors (in Hertz), der Wert ist ein Beispiel
xdepth	Werte: 8 15 16 24	setzt die Farbtiefe, die von X.org benutzt wird (nicht alle Treiber unterstützen 1 und 4)
keytable	z.B. us, de, gb	Tastaturlayout, das von X.org benutzt wird
xkbmodel	(z.B.) pc105	Tastaturtyp, der von X.org benutzt wird (die Zahl bezeichnet die Anzahl der Tasten)

Bootoption	Wert	Beschreibung
xkboptions	(z.B.) grp:alt_shift_toggle	Belegungsvariante der Tastatur, die von X.org benutzt wird
xkbvariant	(z.B.) nodeadkeys,	Setzen einer Belegungsvariante der Tastatur
xmode	800x600	setzt die Bildschirmauflösung nach dem gegebenen Wert (1024x768, 1600x1200 etc.)
xmodule or xdriver	ati, fbdev, i810, intel, mga, nouveau, radeon, savage, vesa	nutzt das gewählte X-Modul
xrandr		erzwingt X.org-Konfiguration unter Verwendung der neuen RandR-1.2-Erweiterungen der X.org-Treiber
xrate	XX	erzwingt eine bevorzugte Wiederholungsfrequenz bei Treibern, die durch RandR 1.2 unterstützt sind. Diese Option muss in Verbindung mit der Bootoption xmode verwendet werden. Eine ausführliche Dokumentation findet sich hier
xhrefresh	75	setzt die horizontale Frequenz des Monitors für X (in Kilohertz), der Wert ist Beispiel
xvrefresh	60	setzt die vertikale Frequenz des Monitors für X (in Hertz), der Wert ist Beispiel

3.3.3 Allgemeine Parameter des Linux-Kernels

Bootoption	Wert	Beschreibung
apm	off	schaltet Advanced Power Managment aus

Bootoption	Wert	Beschreibung
1, 3, 5	(z.B.) 3	Boot-Ziele bzw. Runlevel, die man manuell in der Grub-Bootzeile eingeben kann. Siehe auch die Handbuchseite Runlevel - Ziel-Unit
irqpoll		benutzt IRQ-Polling
mem	(z.b) 128M, 1G	benutzt die angegebene Speichergröße
noagp		keine AGP-Unterstützung (Accelerated Graphics Port)
noapic		keine APIC-Abfrage (Advanced Programmable Interrupt Controller)
nodma		keine Unterstützung für DMA (Direct Memory Access)
noisapnpbios		führt keine ISA-“Plug and Play”-Abfrage beim Start durch
nomce		deaktiviert die Kernel-Option “Machine Check Exception”
nosmp		verwendet keinen Symmetric Multi-Prozessor (mehrere CPUs oder CPUs mit Hyper-Threading)
pci	noacpi	kein ACPI für PCI-Geräte
quiet		es erfolgt keine Ausgabe am Bildschirm
vga	normal	mehr zu vga-Codes im nächsten Absatz
video	(z.B.) DVI-0:800x600	Für Grafikkarten mit aktiviertem KMS. Dies gilt für Intel- und ATI-Grafikkarten (Letztere mit Radeon-Treiber), wobei DVI-X/LVDS-X die Video-Ausgabe ist, die von xrandr gezeigt wird.

3.3.4 VGA-Codes

Die folgenden Tabellen listen die Werte, die mit dem allgemeinen Parameter `vga` angegeben werden können.
Ein Anwendungsbeispiel ist `vga=791` (VESA-Code, Auflösung 1024x768 bei 64000 Farben)

Probleme bei Netbooks oder anderen Bildschirmauflösungen können mit der Eingabe von `vga=0` in die Grubzeile gelöst werden.

Dezimal

Farben	640x480	800x600	1024x768	1280x1024
256	257	259	261	263
32k	272	275	278	281
64k	273	276	279	282
16M	274	277	280	

Hexadezimal

Farben	640x480	800x600	1024x768	1280x1024
256	0x101	0x103	0x105	0x107
32k	0x110	0x113	0x116	0x119
64k	0x111	0x114	0x117	0x11A
16M	0x112	0x115	0x118	

VESA

Farben	640x480	800x600	1024x768	1280x1024	1600x1200
256	769	771	773	775	796
32k	784	787	790	793	797
64k	785	788	791	794	798
16M	786	789	792	795	

Zuletzt bearbeitet: 2021-05-03

3.4 ISO download

Bitte verwende den nächstgelegenen Spiegelserver. Spiegelserver, die unterhalb des Links mit Angaben für den Eintrag in `/etc/apt/sources.list.d/siduction.list` gelistet sind, werden zeitnah aktualisiert.

Europa

- Office Vienna, Wien, Österreich
  https://siduction.office-vienna.at/
- Freie Universität Berlin/spline (Student Project LInux NEtwork), Deutschland
  http://ftp.spline.de/pub/siduction/
  https://ftp.spline.de/pub/siduction/
  ftp://ftp.spline.de/pub/siduction/
- Universität Stuttgart, Deutschland
  https://ftp.uni-stuttgart.de/siduction/
  ftp://ftp.uni-stuttgart.de/siduction/
- Academic Computer Club, Universität Umeå, Schweden
  http://ftp.acc.umu.se/mirror/siduction.org/
  https://ftp.acc.umu.se/mirror/siduction.org/
  rsync://ftp.acc.umu.se/mirror/siduction.org/
- Dotsrc.org, Universität Aalborg, Dänemark
  http://mirrors.dotsrc.org/siduction/
  https://mirrors.dotsrc.org/siduction/
  ftp://mirrors.dotsrc.org/siduction/
  rsync://mirrors.dotsrc.org/siduction/
- Yandex, Moskau, Russland
  http://mirror.yandex.ru/mirrors/siduction/
  https://mirror.yandex.ru/mirrors/siduction/
  ftp://mirror.yandex.ru/mirrors/siduction/
  rsync://mirror.yandex.ru/mirrors/siduction/
- GARR Consortium, Italien
  http://siduction.mirror.garr.it/
  https://siduction.mirror.garr.it/

- Quantum Mirror, Ungarn
  http://quantum-mirror.hu/mirrors/pub/siduction/
  https://quantum-mirror.hu/mirrors/pub/siduction/
  rsync://quantum-mirror.hu/siduction/
- Belnet, Brüssel, Belgien
  http://ftp.belnet.be/mirror/siduction/
  https://ftp.belnet.be/mirror/siduction/
  ftp://ftp.belnet.be/mirror/siduction/
  rsync://ftp.belnet.be/siduction/
- Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen, Deutschland
  https://ftp.gwdg.de/pub/linux/siduction/
  ftp://ftp.gwdg.de/pub/linux/siduction/
  rsync://ftp.gwdg.de/pub/linux/siduction/
- RWTH Aachen, Deutschland
  http://ftp.halifax.rwth-aachen.de/siduction/
  https://ftp.halifax.rwth-aachen.de/siduction/
  ftp://ftp.halifax.rwth-aachen.de/siduction/
  rsync://ftp.halifax.rwth-aachen.de/siduction/
- Studenten Net Twente, Niederlande
  http://ftp.snt.utwente.nl/pub/linux/siduction/
  https://ftp.snt.utwente.nl/pub/linux/siduction/
  ftp://ftp.snt.utwente.nl/pub/linux/siduction/
  rsync://ftp.snt.utwente.nl/siduction/

Asien

- KoDDOS, Amarutu Technology, Hongkong
  http://mirror-hk.koddos.net/siduction/
  https://mirror-hk.koddos.net/siduction/
  rsync://mirror-hk.koddos.net/siduction/

Südamerika

- Corporación Ecuatoriana para el Desarrollo de la Investigación y la Academia, Cuenca
  http://mirror.cedia.org.ec/siduction/

https://mirror.cedia.org.ec/siduction/
rsync://mirror.cedia.org.ec/siduction/

Nordamerika

- Department of Mathematics, Princeton University, United States
  http://mirror.math.princeton.edu/pub/siduction/
  https://mirror.math.princeton.edu/pub/siduction/
- Georgia Tech Software Library (GTlib), Atlanta, United States
  http://www.gtlib.gatech.edu/pub/siduction/
  ftp://ftp.gtlib.gatech.edu/pub/siduction/
  rsync://rsync.gtlib.gatech.edu/siduction/
- Liquorix.net, United States
  https://liquorix.net/siduction/

3.4.1 Dateien der siduction-Spiegelserver

Jeder Spiegelserver umfasst folgende Dateien:

siduction-20xx-xx-release-name-window-manager-arch-datetimestamp.arch.manifest
siduction-20xx-xx-release-name-window-manager-arch-datetimestamp.iso
MD5SUM
MD5SUM.gpg
SHA256SUM
SHA256SUM.gpg
SOURCES

`xxx.manifest` Die Datei listet alle Pakete der jeweiligen ISO.
`xxx.iso` ist die für den Download angebotene Abbilddatei.
`xxx.md5` und `xxx.sha256` enthalten die Checksummen der Abbilddatei und dienen der Überprüfung der Integrität der ISO.
`xxx.gpg`-Dateien sind die Signaturdateien, mit denen Checksummendateien (.md5 .sha256) auf Änderungen überprüft werden.
`xxx.sources` enthält die Downloadlinks zu den Quellcodedateien der verwendeten Pakete.

Download-Links und Spiegelserver findet man auch auf siduction.org

Das Tar-Archiv mit den Quellen ist für den interessant, der siduction weitervertreiben will. Hier müssen die Sourcen mit weitergegeben werden, um der Lizenz zu genügen. Weitere Informationen gibt es in dem Tar-Archiv.

Wenn jemand einen FTP-Server mit entsprechendem Traffic zur Verfügung stellen kann, sind wir jederzeit in den siduction-Foren oder im IRC irc.oftc.net:6667 #siduction-de erreichbar.

3.4.2 Integritätsprüfung

md5sum

Eine md5sum ist die Prüfsumme einer Datei. Diese Prüfsumme wird zur Integritätsprüfung der zugehörigen ISO-Abbilddatei benutzt. Die siduction ISO-Abbilddateien und ihre entsprechenden md5sum Dateien werden immer im gleichen Verzeichnis zum Download angeboten. So zum Beispiel:

```
siduction-21.3.0-wintersky-kde-amd64-202112231751.iso
siduction-21.3.0-wintersky-kde-amd64-202112231751.iso.md5
```

Bei der Integritätsprüfung wird für die heruntergeladene ISO-Abbilddatei eine md5sum erstellt und diese mit einer früher von uns erstellten Summe in der Datei mit der Erweiterung `.md5` verglichen. Weichen die Prüfsummen voneinander ab, so wurde die ISO-Abbilddatei verändert oder beschädigt. Dieser Test schützt vor der Verwendung manipulierter ISO-Abbilddateien und erspart gegebenen Falls viel Zeit für die Fehlersuche einer nicht funktionsfähigen DVD.

Unter Linux wechselt man im Terminal in das Verzeichnis, in dem sich sowohl die ISO-Abbilddatei als auch die `.md5`-Datei befinden. Anschließend erhält man die md5sum der ISO-Abbilddatei mit **`md5sum siduction-*.iso`** und den Inhalt der `.md5`-Datei mit **`cat siduction-*.iso.md5`**. Verbindet man beide Befehle miteinander, erfolgt die Ausgabe zum einfachen Vergleich direkt untereinander.

```
$ md5sum siduction-*.iso && cat siduction-*.iso.md5
358369ebc617613e3c58afc1af716827  siduction-21.3.0-wintersky-↙
   kde-amd64-202112231751.iso
358369ebc617613e3c58afc1af716827 *siduction-21.3.0-wintersky-↙
   kde-amd64-202112231751.iso
```

Noch einfacher gestaltet sich die Überprüfung unter Linux mit dem Befehl **`md5sum -c`**. Achtung, dem Aufruf muss die .md5-Datei mitgegeben werden.

```
    (Befehl und Ausgabe bei Erfolg)
$ md5sum -c siduction-21.3.0-wintersky-kde-amd64-202112231751.iso.md5
siduction-21.3.0-wintersky-kde-amd64-202112231751.iso: OK

    (Befehl und Ausgabe bei einem Fehler)
$ md5sum -c siduction-21.3.0-wintersky-kde-amd64-202112231751.iso.md5
siduction-21.3.0-wintersky-kde-amd64-202112231751.iso: FEHLSCHLAG
md5sum: WARNUNG: 1 berechnete Prüfsumme passte NICHT
```

sha256sum

Die Überprüfung mittels sha256sum ist in der Handhabung identisch mit md5sum. Der wesentliche Unterschied besteht in der Sicherheit durch eine 256 Bit große Prüfsumme (md5sum 128 Bit).

Windows

Bei einem Download der siduction ISO-Abbilddatei in Windows besteht ab Windows 7 innerhalb der Powershell die Möglichkeit mit dem vorinstallierten Hilfsprogramm `CertUtil` Prüfsummen zu erstellen. Der Aufruf lautet dort:

```
CertUtil -hashfile C:\TEMP\<mein_ISO_Abbild.img> MD5
    oder
CertUtil -hashfile C:\TEMP\<mein_ISO_Abbild.img> SHA256
```

Für ältere Windows Versionen ist das unter der General Public License veröffentlichte Programm `md5summer` (486 KB) erhältlich.

Zuletzt bearbeitet: 2022-03-27

3.5 ISO auf USB-Stick - Speicherkarte

Nachfolgend beschrieben wir Methoden eine siduction ISO-Abbilddatei als Live-Medium auf einen USB-Stick, eine SD-Karte oder eine SHDC-Karte zu schreiben. Falls *"persist"* benötigt wird, empfohlen wir die Handbuchseite Aus ISO-Datei booten.

Voraussetzungen

- Das BIOS des PC muss das Booten mittels eines USB-Sticks bzw. einer SD-Karte erlauben. Normalerweise ist dies der Fall, wenn im BIOS des PC diese Bootoption angeboten wird.
- Ein USB-Stick oder eine SD-Karte mit einer empfohlenen Kapazität von mindestens 4 GB.
- Sichere zuvor alle deine Daten auf den Geräten die du für die Herstellung des siduction Live-Medium verwenden möchtest. Auch das aktuell benutzte Betriebssystem und deine privaten Daten. Ein kleiner Tippfehler oder ein vorschneller Klick können alle deine Daten zerstören!

Wichtige Information
Die folgenden Methoden werden vorhandene Partitionstabellen auf dem Zielmedium überschreiben, wodurch alle Daten verloren gehen. Achte auf äußerste Sorgfalt bei der Auswahl des Zielmediums und seiner Laufwerksbezeichnung.

3.5.1 GUI Anwendung

Für Linux™, RasPi™, MS Windows™ oder Mac OS X™

Das kleine Tool USBImager ist für alle oben genannten Betriebssysteme verfügbar und dient der Datensicherung und dem Erstellen des Live-Mediums. Das Programm ist Open Source und lizenziert unter der MIT-Lizenz. Lade die für dein verwendetes Betriebssystem notwendige Datei herunter und installiere das Programm entsprechend den Angaben auf der Downloadseite.

Die Handhabung ist Dank der schnörkellosen Oberfläche denkbar einfach.

Image-Datei auf das Gerät schreiben: 1. Wähle ein Image aus, indem du auf `...` in der ersten Zeile klickst. 2. Wählen ein Gerät aus, indem du in die 3. Zeile klickst. 3. Klicke auf die Schaltfläche `Schreiben` in der 2. Zeile.

Ausführliche Informationen bietet das Readme der Projektseite.

3.5.2 Linux Kommandozeile

Wir empfehlen die Verwendung der Kommandozeile. Die Installation zusätzlicher Programme erübrigt sich, da bereits alle benötigten Werkzeuge vorhanden sind. Eine einzige, leicht verständliche Befehlszeile reicht aus um die siduction ISO-Abbilddatei auf das Speichermedium zu übertragen.

Bevor wir die siduction ISO-Abbilddatei auf das Speichermedium schreiben, müssen wir dessen Laufwerksbezeichnung ermitteln. Am einfachsten benutzen wir journald. Der Befehl **`journalctl -f`** in einem Terminal ausgeführt, zeigt fortlaufend die Meldungen des systemd. Nun stecken wir das Speichermedium an und beobachten die Meldungen im Terminal. Zeilen der folgenden Art enthalten die gesuchten Informationen.

```
kernel: usb 2-3.3: new high-speed USB device number 7 using ⇙
   xhci_hcd
[...]
kernel: scsi 1:0:0:0: Direct-Access     Intenso  Alu Line ⇙
           8.07 PQ: 0 ANSI: 4
kernel: sd 1:0:0:0: Attached scsi generic sg1 type 0
kernel: sd 1:0:0:0: [sdb] 7866368 512-byte logical blocks: ⇙
   (4.03 GB/3.75 GiB)
[...]
kernel: sd 1:0:0:0: [sdb] Write cache: disabled, read cache: ⇙
   enabled, doesn't support DPO or FUA
```

Es handelt sich um einen Intenso USB-Stick mit 4 GB Speicherkapazität und einer Sektorengröße von 512 Bytes. Die Laufwerksbezeichnung lautet `sdb`. Daraus folgt, dass `/dev/sdb` der zu verwendende Pfad für das Zielmedium ist.
Angenommen die siduction ISO-Abbilddatei wurde im `/home` Verzeichnis des Benutzers **tux** gespeichert, können wir mit den Kommandos `dd` oder `cat` das Zielmedium beschreiben. Die Kommandos erfordern Root-Rechte. Daher je nach System

entweder **sudo** bzw. **doas** voranstellen oder ein Terminal verwenden und mit **su** zu **root** werden.

```
dd if=/home/tux/siduction-21.3.0-wintersky-kde-amd64⤶
   -202112231751.iso of=/dev/sdb
    (oder)
cat /home/tux/siduction-21.3.0-wintersky-kde-amd64⤶
   -202112231751.iso > /dev/sdb
```

Der Kopiervorgang kann bei einer etwa 3 GB großen ISO-Abbilddatei durchaus 15 Minuten oder länger benötigen. Bitte entspannt abwarten bis die Eingabeaufforderung zurückkehrt.

3.5.2.1 Zusätzliche Datenpartition Meist ist das Speichermedium deutlich größer als die ISO-Abbilddatei. Die bisher gezeigten Methoden verwenden alle das gesamte Speichermedium, obwohl die ISO-Abbilddatei nur 2,9 GiB belegt. Das lässt sich im Nachhinein nicht ändern. Es bietet sich an, die Vorteile der Kommandozeile zu nutzen und vorausschauend zwei Partitionen einzurichten. Die erste Partition beinhaltet später das Live-System und die zweite den sonst ungenutzten Speicherplatz. Dadurch haben wir die Möglichkeit Daten auf dem Medium zur Life-Sitzung mitzunehmen und während der Life-Sitzung dort abzulegen.

Wir benutzen als root den Befehl `cgdisk /dev/sdb` um eine neue GUID-Partitionstabelle zu erstellen (siehe die Handbuchseite Partitionieren mit gdisk) und verwenden folgenden Daten:

1. Partition:
   Startsektor: 64 (Voreinstellung)
   Größe: 3G (3 GB, etwas größer als die ISO-Abbilddatei)
   Typ Hex-Code: 0700 (Microsoft basic data)
   Name: siduction

2. Partition:
   Startsektor: xxxxxxxx (Voreinstellung, 1. Sektor nach der vorherigen Partition)
   Größe: xxxxxxxx (Voreinstellung, die maximal mögliche Größe)
   Typ Hex-Code: 8300 (Linux)
   Name: data

Wir schreiben die Partitionstabelle auf das Medium und beenden `cfdisk`, bleiben aber noch in der root-Konsole, denn die zweite Partition benötigt noch ein Dateisystem und ein aussagekräftiges Label um sie während der Life-Sitzung nach dem Mounten leichter im Dateimanager zu finden. Die Befehle lauten:

```
mkfs.ext4 -L LifeData /dev/sdb2
```

Mit dem so vorbereiteten Speichermedium schreiben wir die ISO-Abbilddatei in die **1. Partition**.

```
dd if=/home/tux/siduction-21.3.0-wintersky-kde-amd64⇙
   -202112231751.iso of=/dev/sdb1
```

Bitte unbedingt auf `/dev/sdb1` achten. Wenn nur `/dev/sdb` benutzt wird überschreibt der dd-Befehl gnadenlos unsere neu erstellte Partitionstabelle.

3.5.3 Mac OS X Kommandozeile

Der Kopiervorgang ähnelt stark dem Vorgehen bei einem Linux Betriebssystem. Schließe dein USB-Gerät an, Mac OS X sollte es automatisch einbinden. Im Terminal (unter `Applications` > `Utilities`), wird dieser Befehl ausgeführt:

```
diskutil list
```

Stelle die Bezeichnung des USB-Gerätes fest und hänge die Partitionen aus (unmount). In unserem Beispiel ist die Bezeichnung `/dev/disk1`:

```
diskutil unmountDisk /dev/disk1
```

Angenommen die siduction ISO-Abbilddatei wurde im `/home` Verzeichnis des Benutzers **steve** gespeichert, und das USB-Gerät hat die Bezeichnung `disk1`, so führt man folgenden Befehl aus:

```
dd if=/Users/steve/siduction-21.3.0-wintersky-kde-amd64⇙
   -202112231751.iso of=/dev/disk1
```

Zuletzt bearbeitet: 2022-03-10

3.6 ISO brennen

Bevor die ISO-Abbilddatei auf eine DVD gebrannt wird, sollte man sie stets mit Hilfe der von siduction immer angebotenen md5sum und sha256sum prüfen. Das erspart unter Umständen viel Zeit für die Fehlersuche bei einer veränderten oder beschädigten Datei.
Eine ausführliche Anleitung steht im Handbuchkapitel ISO Download, Integritätsprüfung.

WICHTIGE INFORMATION:
siduction, als Linux-LIVE-DVD/CD, ist sehr stark komprimiert. Aus diesem Grund muss besonders auf die Brennmethode des Abbildes geachtet werden. Bitte verwende hochwertige Medien, das Brennen im DAO-Modus (disk-at-once) und nicht schneller als achtfach (8x).

Wir empfehlen allerdings, sofern die Hardware das Booten von USB unterstützt, das Abbild auf einen USB-Stick oder eine SD-Speicherkarte zu legen. Eine Anleitung dazu bietet die Handbuchseite ISO auf USB-Stick / Speicherkarte.

3.6.1 DVD mit Linux brennen

Wer bereits Linux auf dem Rechner hat, kann die DVD mit jedem installierten Brennprogramm erstellen. Je nach Desktopumgebung sind das die Programme
+ **`K3b`** für KDE
+ **`Brasero`** für Gnome
+ **`Xfburn`** für XFCE, LXQt und Gnome

Die Brennprogramme sind in ihrer Bedienung weitgehend selbsterklärend.
Bei K3b wählt man `Weitere Aktionen...` -> `Abbild schreiben...` aus.
In Xfburn und Brasero sollte man `Abbild brennen` anklicken.
Anschließend ist das zu brennende ISO-File (z.B. siduction-21.3.0-wintersky-kde-amd64-202112231751.iso) auszuwählen und der Brennmodus `DAO` (Disk At Once) oder `Automatisch` einzustellen und der Brennvorgang zu starten.

Gelegentliche Probleme beim Brennen der Live-DVD haben ihre Ursache zumeist in den graphischen Frontend-Applikationen. Dies kann man umgehen, indem auf der Konsole das sehr einfach anzuwendende Skript `burniso` benutzt wird. Die Handbuchseite DVD ohne GUI brennen erklärt die Verwendung von burniso kurz

und exakt, sowie weitere Befehle um verfügbare Hardware zu erkennen, Daten zusammenzustellen und CD/DVDs zu brennen.

3.6.2 DVD mit Windows brennen

Selbstverständlich kann man die DVD auch in Windows brennen. Die heruntergeladene Datei muss als ISO-Abbild auf eine DVD gebrannt werden und nicht aus dem Windows Explorer heraus als Datei.
Es gibt verschiedene gute Programme, die die mit Windows Vista eingeführte, integrierte Brennfunktion für CD und DVD erweitern um ISO-Dateien zu brennen. Hier nur zwei Beispiele.

- Die Open-Source-Software cdrtfe ist in der aktuellen Version kompatibel mit Windows Vista, 7, 8, 10 und 11. Mit dem Programm lassen sich ISO-Abbilder brennen, Daten-Disks (CD, DVD, BD) sowie Audio und Video CD/DVDs erstellen. Man kann es in Windows installieren oder das zip-Archiv herunterladen und nach dem Entpacken `cdrtfe` ohne weitere Installation starten.

- Die Closed-Source-Software `CDBurnerXP` ist ein kostenloses Programm, dass neben dem Brennen von ISO-Abbildern auch Daten und Audio CD/DVDs erstellen kann und wieder beschreibbare Medien bei Bedarf löscht. Erhältlich unter CDBurnerXP.

Zuletzt bearbeitet: 2022-04-06

3.7 Life-DVD ohne GUI brennen

WICHTIGE INFORMATION:
siduction, als Linux-LIVE-DVD/CD, ist sehr stark komprimiert. Aus diesem Grund muss besonders auf die Brennmethode des ISO-Abbilds geachtet werden. Wir empfehlen hochwertige CD-Medien (oder DVD+R), das Brennen im DAO-Modus (disk-at-once) und nicht schneller als achtfach (8x).

Man benötigt zum Brennen einer CD/DVD nicht notwendigerweise eine grafische Benutzeroberfläche (GUI).
Probleme, die beim Brennen auftreten, haben ihre Ursache normalerweise in den Frontends wie K3b, nicht so häufig in den Backends wie growisofs, wodim oder cdrdao.

Bevor die ISO-Abbilddatei auf eine DVD gebrannt wird, sollte man sie stets mit Hilfe der von siduction immer angebotenen md5sum und sha256sum prüfen. Das erspart unter Umständen viel Zeit für die Fehlersuche bei einer veränderten oder beschädigten Datei.
Eine ausführliche Anleitung steht im Handbuchkapitel ISO Download, Integritätsprüfung.

3.7.1 burniso

Siduction stellt ein Skript namens `burniso` zur Verfügung.
Burniso brennt unter Nutzung von wodim die ISO-Abbilddatei im Disk-At-Once-Modus mit einer fest eingestellten Brenngeschwindigkeit von 8x. Zuvor testet burniso ob die notwendige Hardware verfügbar ist und listet dann alle erkannten ISO-Abbilddateien auf.

Als **user** wechselt man in das Verzeichnis mit den ISO-Abbilddateien und ruft `burniso` auf:

```
$ cd /Pfad/zur/ISO
$ burniso
Using device /dev/sr0.
Choose an ISO to burn:
1) siduction-21.3.0-wintersky-kde-amd64-202112231751.iso
2) siduction-21.3.0-wintersky-lxqt-amd64-202112231805.iso
3) siduction-21.3.0-wintersky-xfce-amd64-202112231826.iso
```

```
#? _
```

Nach Eingabe der Ziffer für die gewünschte ISO-Abbilddatei prüft `burniso` die Integrität sofern im gleichen Verzeichnis eine zugehörige `.md5` Datei liegt. Bei Erfolg startet der Brennvorgang unmittelbar anschließend. Daher soll man darauf achten, dass vor dem Start des Skripts bereits das Medium, auf das gebrannt wird, eingelegt ist.

Burniso perfektioniert und vereinfacht für den Anwender eine einzige Funktion, nämlich das Brennen von ISO-Abbildern. Darüber hinaus bieten die Kommandozeilenprogramme alle Möglichkeiten um Medien mit Daten verschiedenster Art auf CD, DVD und BD zu erstellen. Im folgenden Kapitel zeigen wir einige Beispiele die häufig Anwendung finden.

3.7.2 Brennen mit cdrdao wodim growisofs

Die Kommandozeilenprogramme bilden die Basis für die beliebten GUI-Programme wie `K3b`, `Brasero` oder `Xfburn`. Wer die vollständige Bandbreite an Optionen der Programme `cdrdao`, `wodim`, `growisofs` usw. bevorzugt, nutzt die Kommandozeile. Wir präsentieren hier nur einen minimalen Ausschnitt der Möglichkeiten. Das Studium der Manpages sollte selbstverständlich sein und ist mit den Beispielen etwas einfacher. Darüber hinaus finden sich im Internet mit der Suchmaschine der Wahl Tips für das eigene Projekt.

3.7.3 Verfügbare Geräte

Ist die vorhandene Hardware zum Brennen nicht genau bekannt, analysieren die Programme wodim und cdrdao die Gerätedaten und geben die Informationen aus. Zuerst `wodim` für einen externen DVD-Brenner an USB:

```
$ wodim -checkdrive
Device was not specified. Trying to find an appropriate drive↙
   ...
Detected CD-R drive: /dev/sr0
[...]
Vendor_info    : 'HL-DT-ST'
Identification : 'DVDRAM GP50NB40 '
Revision       : 'RB00'
```

```
Device seems to be: Generic mmc2 DVD-R/DVD-RW.
Using generic SCSI-3/mmc DVD-R(W) driver (mmc_mdvd).
Driver flags   : SWABAUDIO BURNFREE
Supported modes: PACKET SAO
```

Die Ausgabe für das gleiche Gerät mit `cdrdao`:

```
$ cdrdao scanbus
Cdrdao version 1.2.4 - (C) Andreas Mueller
/dev/sr0 : HL-DT-ST, DVDRAM GP50NB40 , RB00
```

Ein weiteres Beispiel mit `wodim` auf einem anderen PC mit zwei IDE/ATAPI Geräten:

```
$ wodim --devices
wodim: Overview of accessible drives (2 found) :
-------------------------------------------------------------
0  dev='/dev/scd0'      rwrw-- : 'AOPEN' 'CD-RW CRW2440'
1  dev='/dev/scd1'      rwrw-- : '_NEC' 'DVD_RW ND-3540A'
-------------------------------------------------------------
```

Wir benötigen für die Kommandozeilenprogramme vor allen Dingen die genaue Bezeichnung der Gerätedatei (*"/dev/sr0"* oder *"/devscd1"*) um den richtigen Brenner zu benutzen.

3.7.4 Beispiele für CD DVD BD

Bei den Beispielen verzichten wir auf umfangreiche Erläuterungen der verwendeten Optionen. Bitte die Manpages für ausführliche Informationen konsultieren.

CD/DVD von einem ISO-Abbild brennen:

Wodim erkennt an Hand der Dateinamenerweiterung `*.iso` und der Option `-dao`, dass ein Abbild zu brennen ist.

```
$ wodim dev=/dev/sr0 driveropts=burnfree,noforcespeed fs=14M ↙
   speed=8 -dao -eject -v <ISO-Abbild.iso>
```

Falls man eine Fehlermeldung zu *“driveropts”* erhält, liegt dies daran, dass burn-free auf einigen Brennern nicht möglich ist. Dies wird gelöst, indem die driveropts aus dem Befehl entfernt werden.

```
$ wodim dev=/dev/sr0 fs=14M speed=8 -dao -eject -v <ISO-Abbild.iso>
```

Mit `genisoimage` und `growisofs` kann man eine ISO-Abbilddatei aus einem Ordner und allen Unterordnern erstellen und anschließend brennen.

```
    (ISO erstellen)
$ genisoimage -o <ISO-Abbild.iso> -r -J -l <directory>
    (ISO brennen)
$ growisofs -dvd-compat -Z /dev/dvd=<ISO-Abbild.iso>
```

Eine CD mittels eines bin/cue-Abbild brennen:

```
$ cdrdao write --speed 24 --device /dev/scd0 --eject filenam.cue
```

CD-RW/DVD-RW löschen

Um wieder beschreibbare Medien mit neuen Daten zu versehen, müssen sie zuvor gelöscht werden. Die Befehle für das Löschen der Inhaltsverzeichnisse lauten:

```
$ wodim -blank=fast -v dev=/dev/sr0
    (oder)
$ cdrdao blank --device /dev/sr0 --blank-mode minimal
```

Will man die gesamten Daten überbrennen, verwenden wir bei wodim `-blank=all` und bei cdrdao `-blank-mode full`.

CD/DVD kopieren

Kopieren wenn nur ein Laufwerk vorhanden ist. Nach dem Einlesen wird das Quellmedium ausgeworfen und man muss den leeren Rohling in das gleiche Laufwerk einlegen um fortzufahren.

```
$ cdrdao copy --fast-toc --device /dev/scd0 --buffers 256 -v2
```

Man kann eine CD “on the fly” kopieren, wenn zwei Laufwerke verfügbar sind.

```
$ cdrdao copy --fast-toc --source-device /dev/scd0 --device /↙
   dev/scd1 --on-the-fly --buffers 256 --eject -v2
```

Audio-CD brennen

Alle wav-Dateien in dem aktuellen Ordner mit 12 facher Geschwindigkeit brennen.

```
$ wodim -v -eject -pad -dao speed=12 dev=/dev/scd0 defpregap↙
   =0 -audio *.wav
```

Dateien auf DVD brennen

```
$ growisofs -Z /dev/dvd -R -J datei1 datei2 datei3 ...
```

Wenn auf der DVD noch Platz ist, kann man mit Hilfe der Option `-M` Dateien hinzufügen.

```
$ growisofs -M /dev/dvd -R -J noch_eine_datei ↙
   und_noch_eine_datei
```

Mit diesem Befehl wird der verbliebene freie Platz auf der DVD mit Nullen gefüllt und das Medium geschlossen.

```
$ growisofs -M /dev/dvd=/dev/zero
```

Zuletzt bearbeitet: 2022-03-08

4 Installation

Dieser Abschnitt beinhaltet Informationen und Hinweise zum/zur

- Installation vom Live-Medium auf HDD, den notwendigen Vorbereitungen, der Partitionierung und eine Anleitung für das Installationsprogramm Calamares.
- Ohne Installation aus ISO-Datei starten
- Partitionierung von Installationsmedien, mit Beispielen verschiedener Plattengrößen und Single- oder Dual-Boot.
- Benennung von Blockgeräten (UUID), den verschiedenen Arten der Benennung, der Verwendung von Labeln, der Anpassung der *fstab* und der Erstellung neuer Einhängepunkte.
- Partitionieren mit dem Programm GParted in der graphischen Oberfläche.
- Partitionieren mit gdisk nach dem UEFI-GPT Standard im Terminal.
- Partitionieren mit fdisk auf Basis des herkömmlichen BIOS mit MBR-Partitionstabellen. (Sollte nur noch bei alter Hardware benutzt werden.)
- LVM-Partitionierung (Logical Volume Manager) - in sechs Schritten zum Ziel, und die Verwaltung von *Logical Volumen*.
- Private Daten aus dem Verzeichnis /home verschieben, um z. B. bei parallelen Installationen eine Daten-Partition für mehrere Betriebssysteme verfügbar zu machen.

Zuletzt bearbeitet: 2022-03-11

4.1 Installation auf HDD

4.1.1 Datensicherung

WICHTIG: IMMER EINE DATENSICHERUNG ANLEGEN!
Wenn auf dem Installationsziel bereits ein Betriebssystem beheimatet ist, oder Daten erhalten bleiben sollen, bitte vor der Installation von siduction immer eine Sicherung anlegen.

4.1.2 Installationsvorbereitungen

Zuerst stellt man die Bootreihenfolge auf das zu bootende Medium (DVD, Flashcard oder USB-Stick) um. Bei den meisten Computern kommt man durch Drücken der **F2** oder **Entf**-Taste während des Bootvorgangs in das Setup von UEFI oder BIOS. Alternativ kann während des Bootvorgangs die Taste **F12**, **F11** **F7** oder **F8** (je nach Angaben der Hardwarehersteller) gedrückt werden um dann das Live-Medium als Startlaufwerk auszuwählen.

siduction startet jetzt in der Regel problemlos. Sollte das nicht der Fall sein, helfen Bootoptionen (Cheatcodes), die an den Bootmanager übergeben werden können. Die Handbuchseite Cheatcodes erläutert die möglichen Optionen.
Am Startbildschirm des Live-Mediums wird, je nachdem was zutrifft, mit den Pfeiltasten zu “From CD/DVD/ISO: …” oder “From Stick/HDD: …” navigiert und die Taste **e** betätigt. So gelangt man zum editieren der Kernelbefehlszeile um die Cheatcodes hinzuzufügen. Mit der Taste **F10** wird der Bootvorgang fortgesetzt.

Vor der Installation bitte alle USB-Sticks, Kameras etc. entfernen.

Soll siduction nicht von, sondern **auf** ein USB-Medium installiert werden, ist ein anderes Verfahren notwendig. Siehe dazu die Handbuchseite Installation auf ein USB-Medium.

HDD, RAM und Swap

Die Mindestanforderungen zur Installation der siduction Varianten sind auf der Handbuchseite Inhalt der Live-ISO beschrieben.
Mit 15 GB Festplattenvolumen und 2 GB Arbeitsspeicher ist man zur Zeit noch auf der sicheren Seite. Auf PCs mit maximal 1 GB RAM sollte eine Swap-Partition

angelegt werden. Mehr als 2 GB Swap wird normal nicht benötigt und ist nur bei Suspend-to-Disk und Serversystemen wirklich sinnvoll.

4.1.3 Partitionierung

Die Partitionierung der Laufwerke ist von vielen Faktoren abhängig:

- Auswahl der siduction-Variante
- Größe der vorhandenen Laufwerke und des Arbeitsspeichers
- Single-Boot oder Dual-Boot mit einem bereits installierten System (Windows, Linux, MAC)
- Gemeinsame Nutzung von Daten für die installierten Systeme

Beispiele und Größen für unterschiedliche Installationssituationen beschreibt die Handbuchseite Partitionierung.
Wir empfehlen, das `/home`-Verzeichnis auf der Wurzel-Partition zu belassen. Das Verzeichnis `/home` sollte der Ort sein, an dem die individuellen Konfigurationen abgelegt werden, und nur diese. Für alle weiteren privaten Daten, dazu zählen auch .ssh, .gnupg und die Mail-Archive, sollte eine eigene Datenpartition angelegt, und falls erforderlich auf das `/home`-Verzeichnis verlinkt werden. Die Vorteile für die Datenstabilität, Datensicherung und auch im Falle einer Datenrettung sind nahezu unermesslich.

Die Partitionierung kann während der Installation vorgenommen werden, oder bereits im Vorfeld während der Live-Sitzung mit den folgenden Programmen:
Gparted, ein Programm für die graphische Oberfläche für GTK-Desktops
KDE Partition Manager, ein weiteres Programm für die graphische Oberfläche für Qt-Desktops
gdisk, empfohlen bei UEFI Hardware für GPT Partitionstabellen
cfdisk, nur für ältere Hardware mit traditionellem BIOS und MBR Partitionstabellen.

4.1.4 Dateisysteme

Wir empfehlen das Dateisystem **ext4**, welches bei siduction als Default-Dateisystem verwendet wird. Dies gilt für alle Partitionen, wenn ausschließlich Linux Betriebssysteme verwendet werden.

Bei einer Dual-Boot Installation mit *Windows* ist eine eigene Datenpartition mit dem **NTFS** Dateisystem sinnvoll. Linux kann lesend und schreibend darauf zugreifen; für Windows ist es das Standarddateisystem.

Bei einer Dual-Boot Installation mit *MAC* ist ebenfalls eine eigene Datenpartition allerdings mit dem **HFS** oder **HFS+** Dateisystem sinnvoll. Linux und MAC können lesend und schreibend darauf zugreifen.

4.1.5 Duplizierung auf einen anderen Computer

Mit folgendem Konsolebefehl wird eine Liste der installierten Softwarepakete erstellt, um mit Hilfe dieser eine identische Softwareauswahl auf einem anderen Computer oder bei einer allfälligen Neuinstallation installieren zu können:

```
~# dpkg -l|awk '/^ii/{ print $2 }'|grep -v -e ^lib -e -dev -e↙
    $(uname -r) >/home/username/installed.txt
```

Am besten wird diese Textdatei auf einen USB-Stick oder einen Datenträger nach Wahl kopiert.
Auf der Zielinstallation wird die Textdatei nach $HOME kopiert und als Referenz verwendet, um die benötigten Programmpakete zu installieren. Die gesamte Paketliste kann per

```
~# apt install $(/home/username/installed.txt)
```

installiert werden.

4.1.6 Das Installationsprogramm Calamares

Während der Installation sollte, wenn möglich, der Computer mit dem Internet verbunden sein, weil Calamares den GeoIP Service verwendet um Voreinstellungen für die Lokalisation und Zeit zu ermitteln.

1. Das Installationsprogramm startet man bequem über das Icon am Desktop oder im Menü: *“System” > “System installieren”*.
2. Nach einem Doppelklick auf das Icon startet Calamares und wir sehen das “Willkommen” - Fenster.

Abbildung 1: calamares welcome

Sofern eine Internetverbindung besteht, sollte hier bereits die richtige Sprache eingestellt sein.

3. Im nächsten Fenster “Standort” besteht die Möglichkeit Änderungen zur *“Region”*, der *“Zeitzone”* und *“Systemsprache”*, sowie dem *“Format”* für das Datum und die Zahlen vorzunehmen.

4. Es folgen die Einstellungen zur Tastatur.

   Im oberen Teil wird die Tastatur graphisch dargestellt und die Änderungen werden sofort sichtbar. Ganz unten befindet sich eine Eingabezeile um das Tastaturlayout zu testen.

5. Im nächsten Schritt erreichen wir die bereits oben erwähnte Partitionierung mit der bestimmt wird, welche Teile der Festplatte(n) siduction verwendet.

   In unserem Beispiel verwenden wir die *“Manuelle Partitionierung”* weil bereits im Vorfeld die Partitionen angelegt wurden und wir nur noch das richtige Installationsziel auswählen. Nach einem Klick auf `Weiter` erscheint das nächste Fenster, in dem wir die einzelnen Partitionen auswählen und bearbeiten können.

   Wir benutzen die Partitionen
   `sda7` für `/` (root)

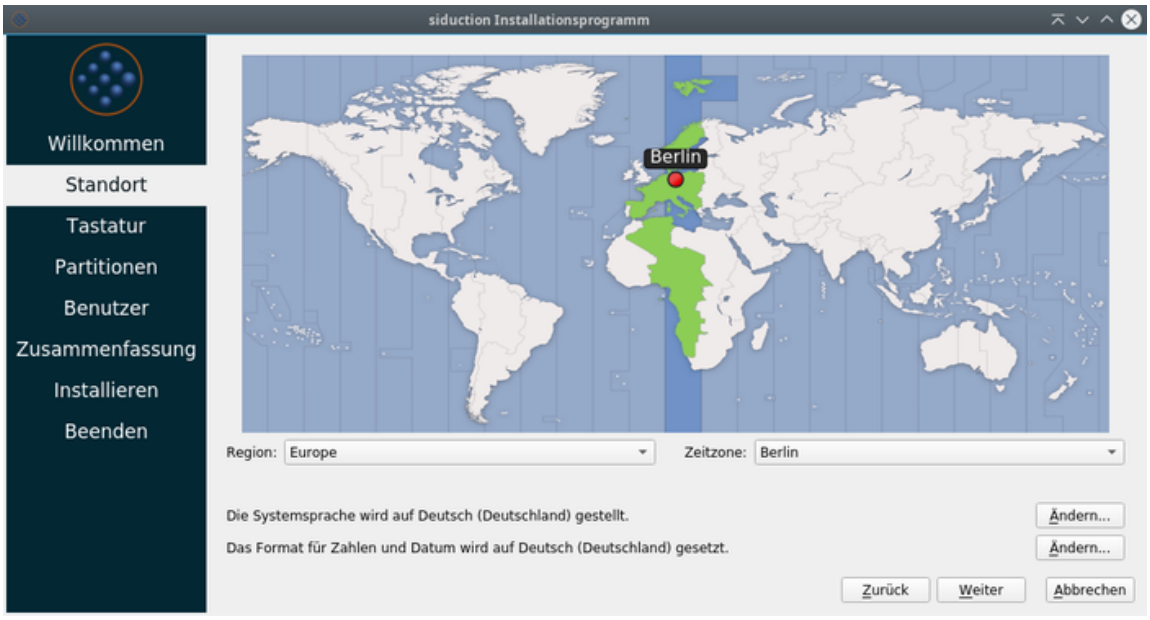


Abbildung 2: calamares location

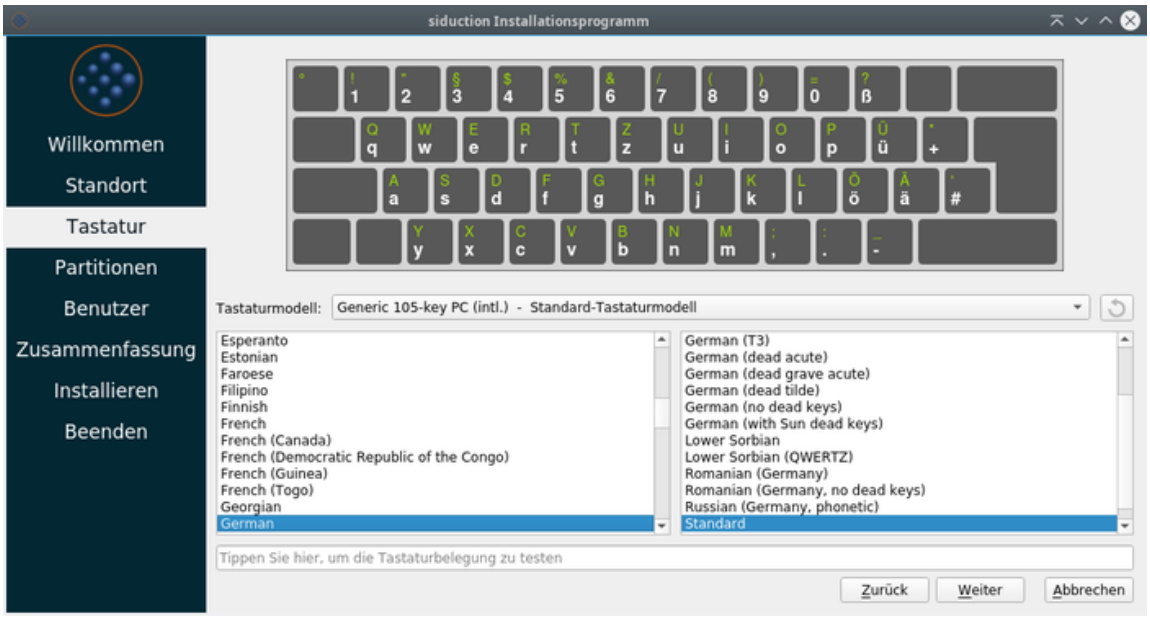


Abbildung 3: calamares keyboard

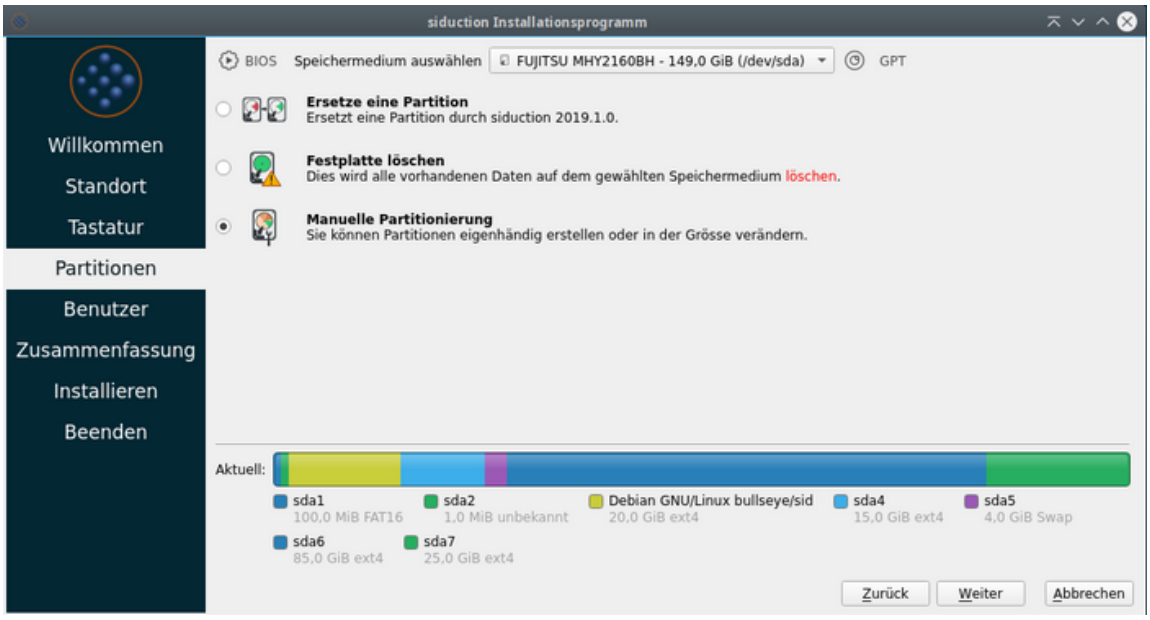


Abbildung 4: calamares partitions

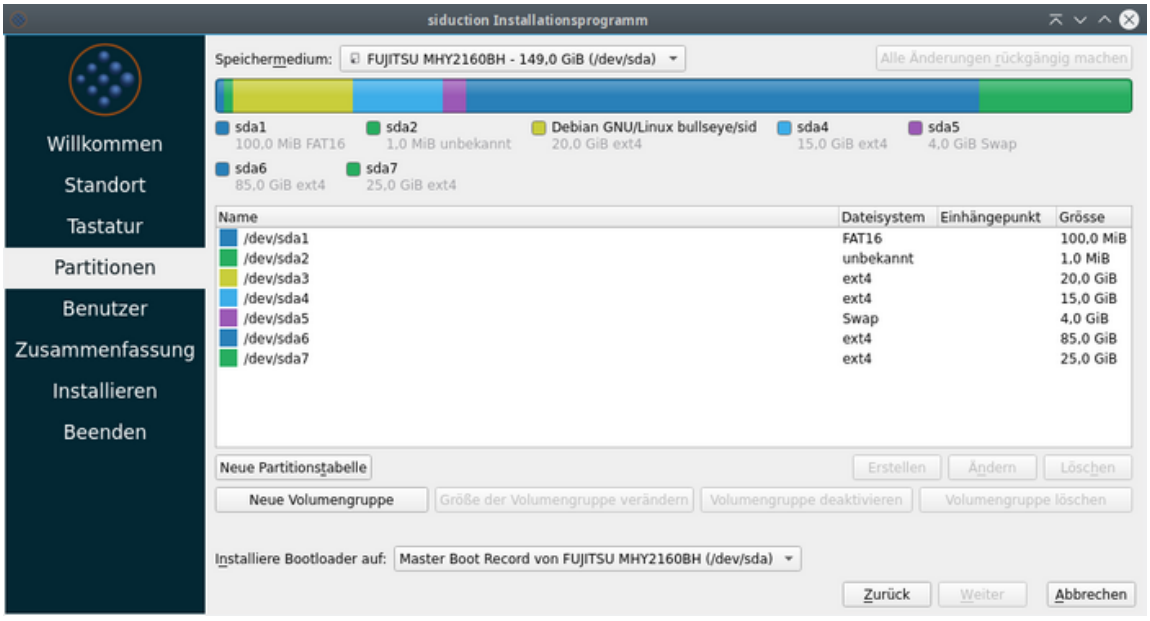


Abbildung 5: calamares work on partitions

`sda6` für `/daten` gemeinsam mit dem bereits auf `sda3` und `sda4` vorhandenem Linux

Nach Auswählen der betreffenden Partition und Betätigen des Schalters `Ändern` öffnet sich ein Fenster, in dem wir den oben bezeichneten Mountpiont eintragen und für `sda7` auch die Formatierung mit dem Dateisystem **ext4** vornehmen. Die Partition `sda6` wird nicht formatiert, da wir die dort schon abgelegten Daten gemeinsam mit dem bereits vorhandenen Linux nutzen möchten.
Die Swap-Partition `sda5` brauchen wir nicht bearbeiten, da sie während der Installation automatisch erkannt und integriert wird.
Das Ergebnis unserer Bemühungen sehen wir im nächsten Bild.

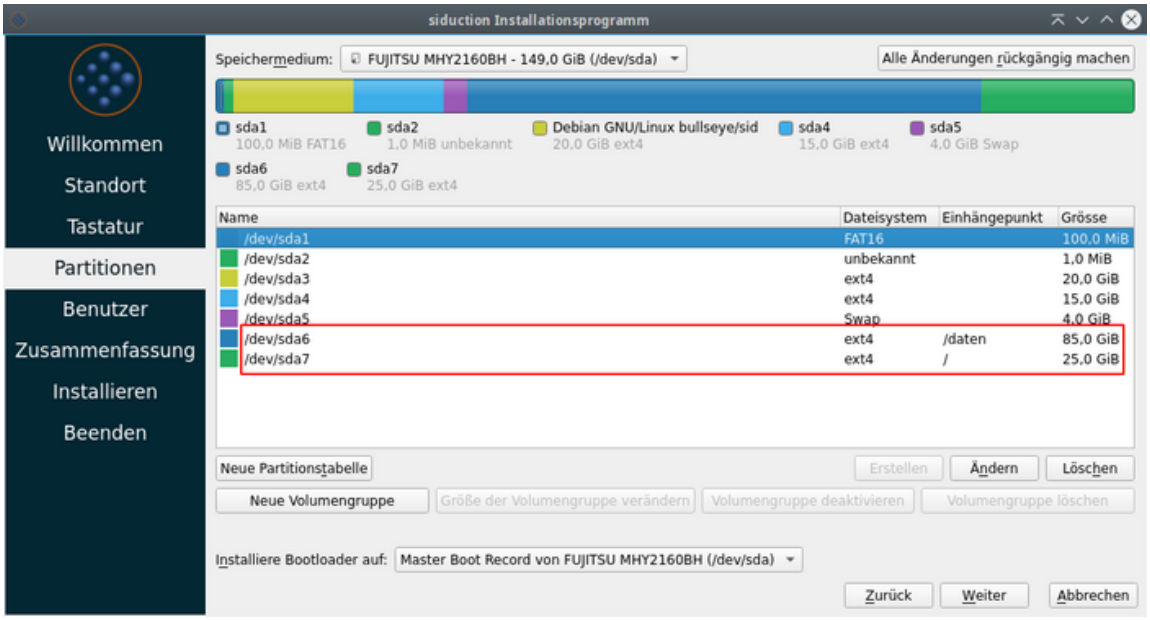


Abbildung 6: calamares partitions finish

6. Als nächstes werden Benutzername, Anmeldename, Computername, Benutzerpasswort und Root-Passwort festgelegt (bitte gut merken!). Die Passwörter sollen aus Sicherheitsgründen nicht zu einfach gewählt werden. Weitere Benutzer können nach der Installation in einem Terminal mit adduser hinzugefügt werden.

   Vor der Verwendung der beiden Optionen
   “Automatisches Einloggen ohne Passwortabfrage” und
   “Nutze das gleiche Passwort auch für das Administratorenkonto”

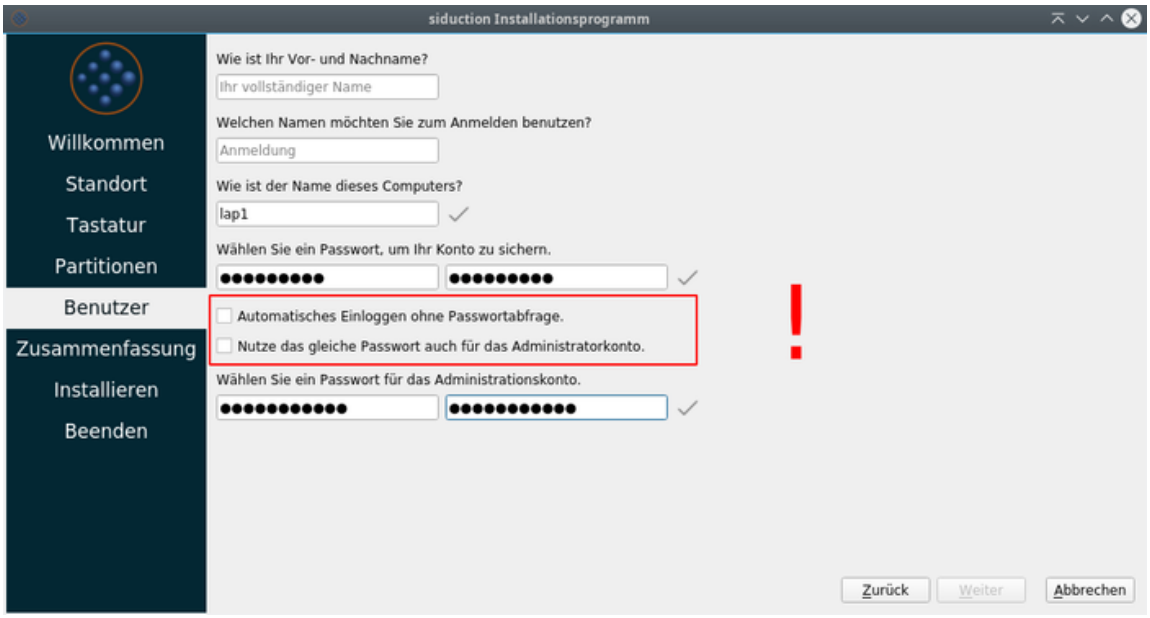


Abbildung 7: calamares users

wird hier ausdrücklich gewarnt. Sie stellen schon für sich allein ein Sicherheitsrisiko dar (siehe auch sudo). Sind beide Optionen aktiviert ist die Eingabe von Passwörtern nur noch eine Farce!

7. Nach Betätigen der Taste `Weiter` erscheint eine Zusammenfassung aller zuvor getätigten Eingaben. Jetzt besteht noch die Möglichkeit über `Zurück` Änderungen vorzunehmen. Sind wir mit dem Ergebnis zufrieden, öffnet ein Klick auf `Installieren` das kleine Warnfenster in dem wir die Installation bestätigen müssen.

8. Nun startet die Installation. Dies dauert je nach Hardware einige Zeit. Der Fortschritt wird entsprechend angezeigt. Auch wenn es etwas länger dauert, bitte die Installation nicht abbrechen, sondern dem Prozess Zeit geben.

9. Am Ende erhalten wir die Möglichkeit zu einem Reboot in das neu installierte System.

   Vor dem Reboot die CD aus dem Laufwerk nehmen!

4.1.7 System verschlüsseln

Ab *siduction 2021.2 Farewell* besteht innerhalb des Installationsprogramms Calamares die Möglichkeit, das vollständige System in eine verschlüsselte Partition

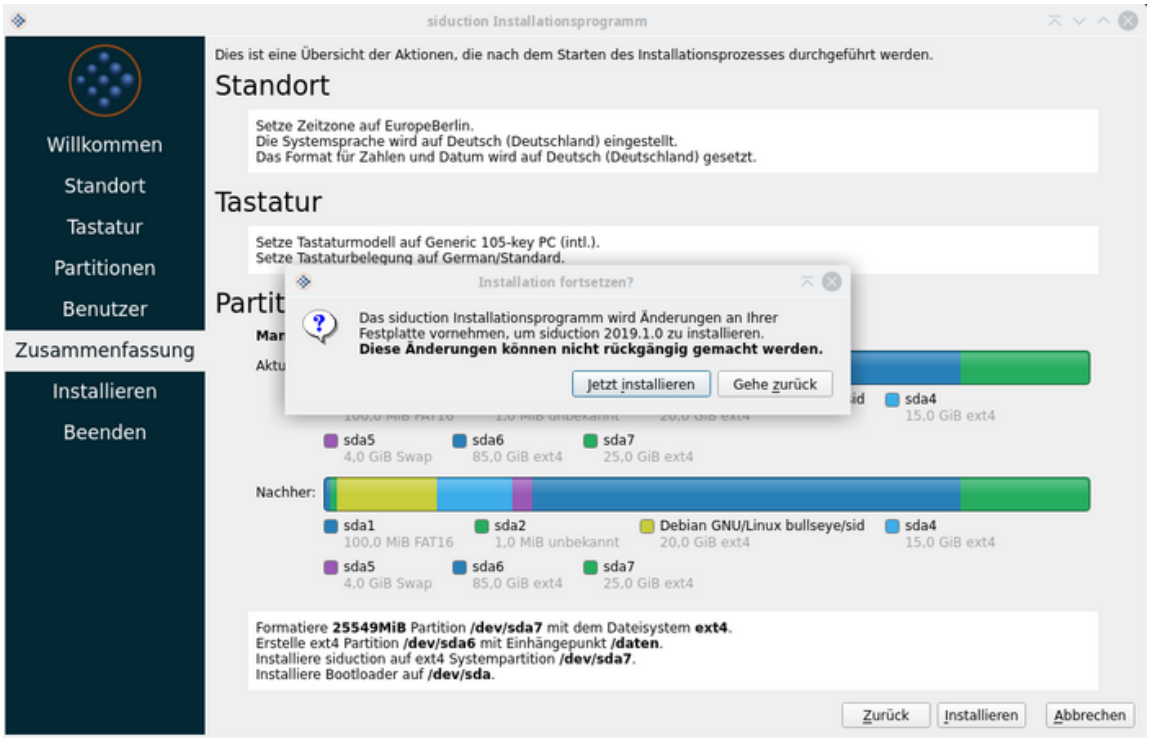


Abbildung 8: calamares summary

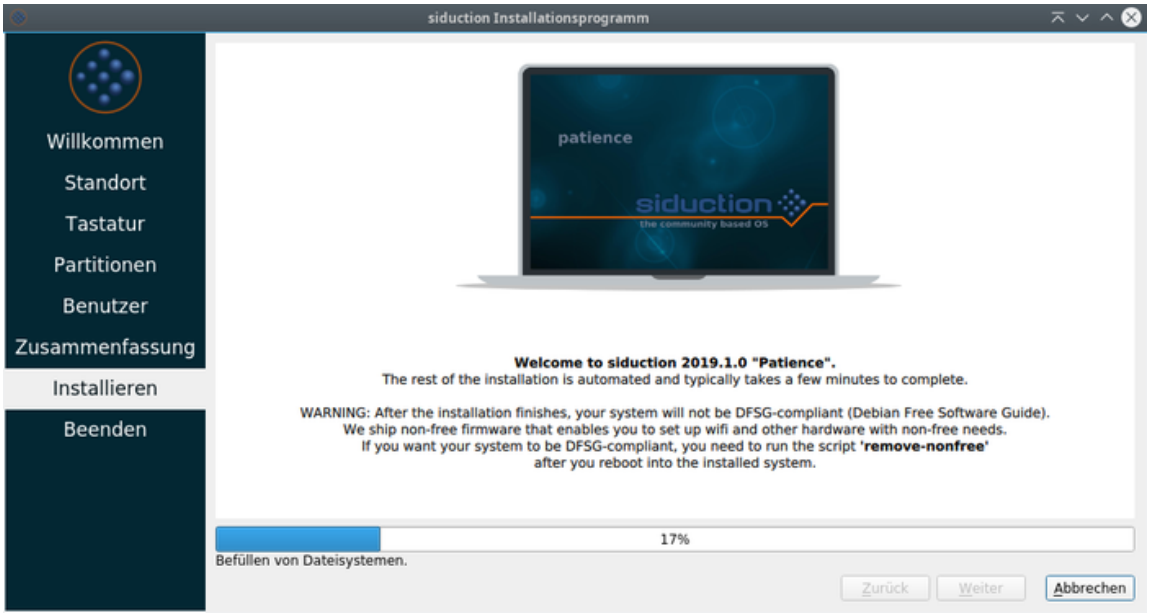


Abbildung 9: calamares install

Abbildung 10: calamares reboot

oder Festplatte zu installieren. Lediglich die erste Stufe des Bootmanagers Grub befindet sich unverschlüsselt auf der "*BIOS-boot*"-Partition. Grub fragt bei jedem Bootvorgang das Passwort für die verschlüsselte Installation ab, bevor das Bootmenü erscheint.
Die oben unter 5 beschriebene Partitionierung gestaltet sich nun geringfügig anders. Bei der Auswahl einer der drei ersten Optionen, erscheint zusätzlich die Funktion "*Verschlüssele System*", in der wir auch gleich das Passwort eingeben.

Benutzen wir "*Manuelle Partitionierung*", so benötigen wir einen leeren, unbenutzten Bereich auf der Festplatte. Ist dieser nicht vorhanden, löschen wir zuerst eine oder mehrere nicht mehr benötigte Partitionen. Dann erstellen wir die neue Partition.

Im nächsten Schritt ist jetzt die Funktion *"Verschlüsseln"* auswählbar.

Wir geben unser Passwort ein und wählen anschließend als Einhängepunkt das Wurzelverzeichnis `/` aus.
Nach Beendigung der Partitionierung setzen wir die Installation mit dem Menüpunkt *"Benutzer"*, wie oben unter 6 beschrieben, fort.

Abbildung 11: calamares encrypt system

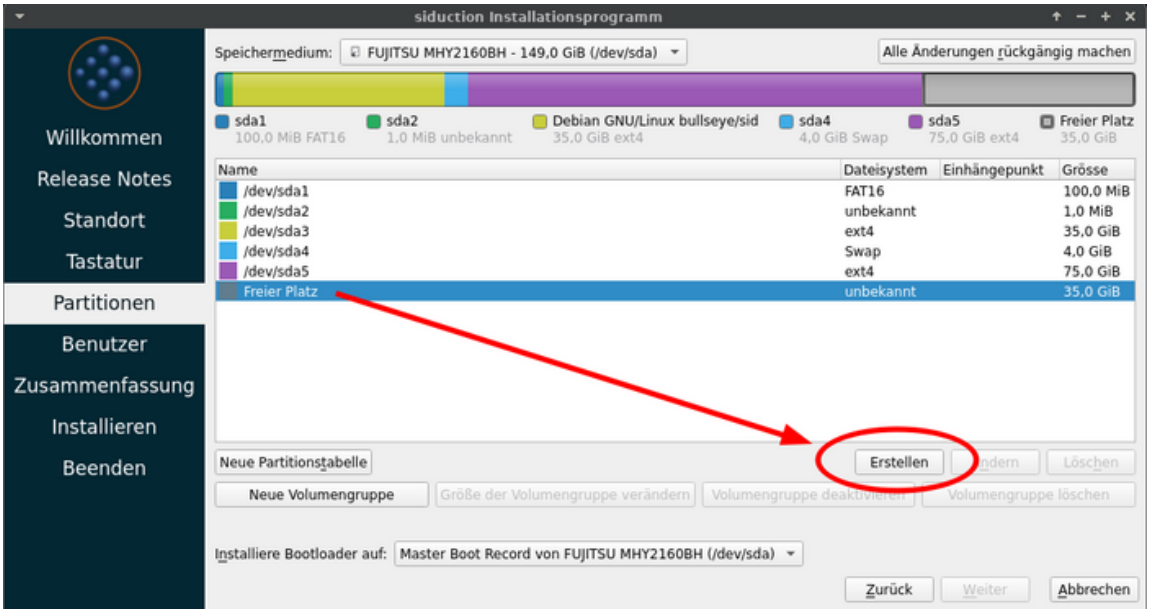


Abbildung 12: calamares, manual partitioning encrypt 1

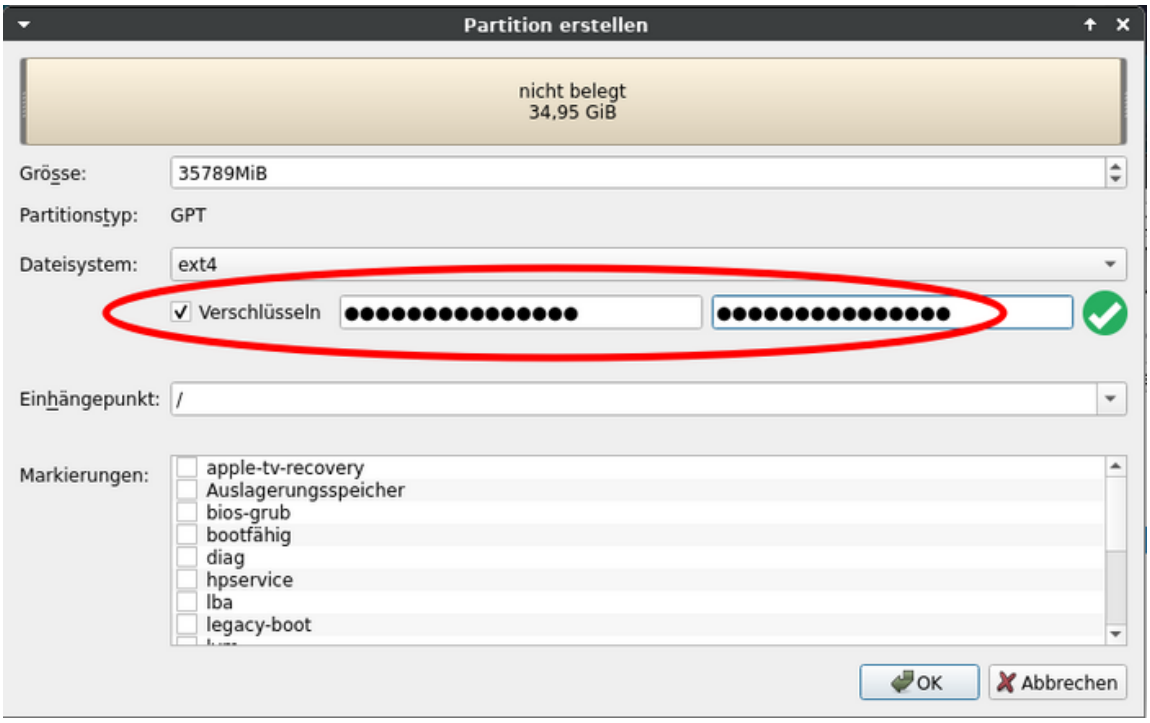


Abbildung 13: calamares, manual partitioning encrypt 2

4.1.8 Benutzer hinzufügen

Um neue Benutzer mit automatischer Übernahme der Gruppenberechtigungen hinzuzufügen, führt man folgenden Befehl als **root** aus:

```
~# adduser <nutzername>
```

Nach drücken der Eingabetaste werden weitere Optionen angezeigt, die zusätzliche Einstellungen ermöglichen. Zuletzt folgt eine Aufforderung zum zweimaligen Eingeben des Passworts.

siduction spezifische Desktopsymbole (für das Handbuch und den IRC) müssen selbst hinzugefügt werden.

So entfernt man einen Benutzer

```
~# deluser <nutzername>
```

Mehr Informationen bieten die Manpages **`man adduser`** und **`man deluser`** .

Zuletzt bearbeitet: 2022-02-09

4.2 Aus ISO-Datei booten

4.2.1 Überblick

Dieser Cheatcode startet aus einer ISO-Datei auf der Festplatte mit dem Dateisystem **ext4**. Für normalen Gebrauch empfehlen wir das Standarddateisystem von siduction, **ext4**, welches von den Maintainern gut betreut ist.

Der Start von einer *"fromiso"* Festplatten-Installationen dauert nur einen Bruchteil der Zeit, die ein Start von einer CD benötigt. Außerdem steht gleichzeitig das CD/DVD-Laufwerk zur Verfügung. Alternativ kann man auch VBox, KVM oder QEMU verwenden.

Voraussetzungen

- eine funktionierende Grub-Installation (auf Floppy, einer Festplatteninstallation oder der Live-CD)

- eine siduction-Imagedatei, z. B. siduction.iso (Name gekürzt) und ein Linux-Dateisystem wie **ext4**

4.2.2 fromiso mit Grub2

siduction liefert eine grub2-Datei mit der Bezeichnung `60_fll-fromiso`, um einen fromiso-Eintrag im grub2-Menü zu generieren. Die Konfigurationsdatei für fromiso ist im Paket `grub2-fll-fromiso` , mit dem Pfad `/etc/default/grub2-fll-fromiso` .

Als erstes öffnet man ein Terminal und wird **root**. Dann istalliert man die Konfigurationsdatei:

```
su
apt update
apt install grub2-fll-fromiso
```

Im Anschluss öffnet man einen Editor der Wahl (kwrite, mcedit, vim …):

```
mcedit /etc/default/grub2-fll-fromiso
```

In den Zeilen, die aktiv sein sollen, wird das Kommentarzeichen # am Anfang der Zeile entfernt, und man ersetzt die voreingestellten Anweisungen innerhalb der doppelten Anführungszeichen mit den eigenen Parametern.

Beispiel: vergleiche diese geänderte grub2-fll-fromiso mit den Grundeinstellungen:

```
# Defaults for grub2-fll-fromiso update-grub helper
# sourced by grub2's update-grub
# installed at /etc/default/grub2-fll-fromiso by the ⇙
   maintainer scripts

#
# This is a POSIX shell fragment
#

# specify where to look for the ISO
# default: /srv/ISO
## Achtung: Dies ist der Pfad zum Verzeichnis, in dem das ⇙
   oder die ISO(s) liegen,
## der Pfad soll das eigentliche siduction.iso nicht ⇙
   inkludieren.
FLL_GRUB2_ISO_LOCATION="/media/disk1part4"

# array for defining ISO prefices --> siduction-*.iso
# default: "siduction- fullstory-"
FLL_GRUB2_ISO_PREFIX="siduction-"

# set default language
# default: en_US
FLL_GRUB2_LANG="de_DE"

# override the default timezone.
# default: UTC
FLL_GRUB2_TZ="Europe/Berlin"

# kernel framebuffer resolution, see
# http://manual.siduction.org/de/cheatcodes-vga-de.htm#vga
# default: 791
#FLL_GRUB2_VGA="791"
```

```
# additional cheatcodes
# default: noeject
FLL_GRUB2_CHEATCODE="noeject nointro"
```

Speichere die Änderungen, schließe den Editor und führe als **root** folgenden Befehl in einem Terminal aus:

```
update-grub
```

Die Grub2-Konfigurationsdatei `grub.cfg` wird damit aktualisiert und erkennt die im angegebenen Verzeichnis platzierten ISOs. Diese stehen beim nächsten Neustart zur Wahl.

4.2.3 toram

Eine weitere Nützliche Option beim Booten von einem Live Medium ist `toram`. Selbige ist empfehlenswert, wenn der Rechner über ausreichend Arbeitsspeicher verfügt (4GiB oder mehr). Damit wird der komplette Inhalt des Live Mediums in den RAM kopiert. Das hat den Vorteil, dass das System schneller reagiert. Außerdem kann man das Medium nach dem Start entfernen, um den USB Port oder das Laufwerk anderweitig zu nutzen.

Zuletzt bearbeitet: 2022-04-14

4.3 Partitionierung von Installationsmedien

Linux-Einsteigern empfehlen wir, nur zwei Partitionen anzulegen `/root` (inkl. `/home`) und `swap`, da dies eine Erstinstallation wesentlich vereinfacht. Nach der Installation können weitere Datenpartitionen angelegt werden, oder etwa ein separates `/home`, falls gewünscht.

Wir raten aber eher davon ab eine `/home`-Partition anzulegen.
Das Verzeichnis `/home` sollte der Ort sein, an dem die individuellen Konfigurationen abgelegt werden, und nur diese. Für alle weiteren privaten Daten sollte eine eigene Datenpartition angelegt werden. Die Vorteile für die Datenstabilität, Datensicherung und auch im Falle einer Datenrettung sind nahezu unermesslich.

Eine `swap`-Partition entspricht in der Funktionalität etwa der Auslagerungsdatei bei Windows, ist aber weit effektiver als diese. Als Faustregel sollte die Swap-Partition zweimal so groß sein wie das verwendete RAM. Dies gilt hauptsächlich für Notebooks, die per `hibernate` in den Ruhezustand versetzt werden sollen, oder Desktops mit sehr wenig RAM (1 GByte oder weniger). Geräte mit ausreichend RAM brauchen heute keine Swap-Partition mehr.

Für den Datenaustausch mit einer Windows-Installation sollte die dafür vorgesehene Partition mit **ntfs** formatiert werden. Siduction kann mit dem automatisch installierten *“ntfs-3g”* lesend und schreibend auf die Daten zugreifen.

Es gibt sehr viele gute Möglichkeiten seine Platten aufzuteilen. Diese Beispiele sollten einen ersten Einblick in die Möglichkeiten bieten.

Die Anschaffung einer externen USB-Festplatte zur regelmäßigen Datensicherung ist ebenso eine Überlegung wert.

4.3.1 Mindestanforderungen

Die Mindestanforderungen für den sinnvollen Gebrauch einer siduction Installation betragen:

Installationssystem	Festplattenplatz
siduction NOX	5GB
siduction Xorg	10GB
siduction LXQt	15GB

Installationssystem	Festplattenplatz
siduction LXde	15GB
siduction XFCE	15GB
siduction Cinnamon	15GB
siduction KDE Plasma	15GB

4.3.2 Beispiele mit verschiedenen Plattengrößen

Falls ein Dual-Boot mit MS Windows™ angelegt wird, muss MS Windows immer als erstes System auf die Festplatte installiert werden.

Als Partitionstabelle sollte der Typ *“GPT”* gewählt werden. So kann man die Vorteile gegenüber *“MBR”* nutzen. Nur bei alter Hardware ist *“MBR”* noch sinnvoll. Die Erklärungen hierzu enthält unsere Handbuchseite Partitionieren mit gdisk.

Die Beispiele beziehen sich auf Partitionstabellen vom Typ *“GPT”*, für deren Funktion die ersten beiden, sehr kleinen Partitionen erforderlich sind.

Dual-Boot mit MS Windows und Linux

1 TB Festplatte:

Partition	Size	Filesystem	Verwendung
1	100 KB	FAT16	EFI-System
2	1 MB	ohne	BIOS-boot
3	50 GB	NTFS	MS Windows System
4	300 GB	NTFS	Daten für MS Windows
5	200 GB	NTFS	Daten für MS Windows und Linux
6	30 GB	ext4	/ (Linux root)
7	416 GB	ext4	Daten für Linux
8	4 GB	Linux Swap	Linux Swap

120 GB Festplatte:

Partition	Größe	Formatierung	Verwendung
1	100 KB	FAT16	EFI-System
2	1 MB	ohne	BIOS-boot

Partition	Größe	Formatierung	Verwendung
3	40 GB	NTFS	MS Windows System
4	48 GB	NTFS	Daten für MS Windows und Linux
5	30 GB	ext4	/ (Linux root)
6	2 GB	Linux Swap	Linux Swap

80 GB Festplatte:

Partition	Größe	Formatierung	Verwendung
1	100 KB	FAT16	EFI-System
2	1 MB	ohne	BIOS-boot
3	40 GB	NTFS	MS Windows System
4	10 GB	NTFS	Daten für MS Windows und Linux
5	28 GB	ext4	/ (Linux root)
6	2 GB	Linux Swap	Linux Swap

Linux allein

500 GB Festplatte:

Partition	Größe	Formatierung	Verwendung
1	100 KB	FAT16	EFI-System
2	1 MB	ohne	BIOS-boot
3	30 GB	ext4	/
4	250 GB	ext4	Daten_1
5	216 GB	ext4	Daten_2
6	4 GB	Linux Swap	Linux Swap

160 GB Festplatte:

Partition	Größe	Formatierung	Verwendung
1	100 KB	FAT16	EFI-System
2	1 MB	ohne	BIOS-boot
1	26 GB	ext4	/

Partition	Größe	Formatierung	Verwendung
3	130 GB	ext4	Daten
4	4 GB	Linux Swap	Linux Swap

60 GB Festplatte:

Partition	Größe	Formatierung	Verwendung
1	100 KB	FAT16	EFI-System
2	1 MB	ohne	BIOS-boot
3	25 GB	ext4	/
4	33 GB	ext4	Daten
5	2 GB	Linux Swap	Linux Swap

4.3.3 Partitionierungsprogramme

- **GParted** Ein einfach zu bedienendes Partitionierungsprogramm mit graphischer Oberfläche.
  Gparted ist auf allen mit einer graphischen Oberfläche ausgestatteten siduction Installationen und Installationsmedien verfügbar. Gparted unterstützt eine Reihe verschiedener Typen von Partitionstabellen. Die Handbuchseite Partitionieren der Festplatte mit GParted liefert weitere Informationen zum Programm.

- **KDE Partition Manager** Ein Qt basiertes, einfach zu bedienendes Partitionierungsprogramm mit graphischer Oberfläche.
  Der KDE Partition Manager ist das Standard-Partitionierungsprogramm für den KDE Destktop, einfach zu bedienen und genauso umfangreich wie Gparted.

- **gdisk / cgdisk** Ein Konsolenprogramm für Partitionstabellen vom Typ *“GPT-UEFI”*.
  gdisk ist das klassische Textmodus-Programm. cgdisk hat eine benutzerfreundlichere ncurses-Oberfläche. Die Handbuchseite Partitionieren mit gdisk liefert weitere Informationen zum Programm.

- **fdisk / cfdisk** Ein Konsolenprogramm für Partitionstabellen vom Typ *“msdos-MBR”*.

Hinweis: fdisk sollte nur noch für alte Hardware, die *"GPT-UEFI"* nicht unterstützt verwendet werden.
fdisk ist das klassische Textmodus-Programm. cfdisk hat eine benutzerfreundlichere ncurses-Oberfläche. Die Handbuchseite Partitionieren mit Cfdisk liefert weitere Informationen zum Programm.

Achtung
Bei Verwendung jedweder Partitionierungssoftware droht Datenverlust. Daten, die noch gebraucht werden, immer zuvor auf einem anderen Datenträger sichern.

Eingebundene Partitionen (auch swap) müssen vor Bearbeitung gelöst werden. Im Terminal als **root** mit dem Befehl:

```
# umount /dev/sda1
```

Die Einbindung einer Swap-Partition wird mit diesem Befehl gelöst:

```
# swapoff -a
```

4.3.4 Weiterführende Infos

Hier die umfassende englischsprachige Dokumentation von GParted

Für weitere Partitionierungsoptionen siehe:

- Logical Volume Manager LVM-Partitionierung
- Partitionierung mit GPT zur Unterstützung von UEFI Partitionieren mit gdisk (GPT fdisk)

Zuletzt bearbeitet: 2021-07-21

4.4 UUID - Benennung von Blockgeräten

UUID (Universally Unique Identifier) und Partitions-Label

Die dauerhafte Benennung (persistent naming) von Blockgeräten wurde mit Einführung von udev ermöglicht. Der Vorteil ist die Unabhängigkeit von den verwendeten Controllern, sowie der Art und der Anzahl der angeschlossenen Geräte. Die bei der Installation von siduction erstellte Datei `/etc/fstab` enthält entsprechende Einträge für alle zu diesem Zeitpunkt angeschlossenen Blockgeräte.

4.4.1 Arten der Benennung von Blockgeräten

Zur Zeit werden in Linux fünf Arten von Bezeichnern für Blockgeräte verwendet. Alle Bezeichner sind unterhalb des Verzeichnisses `/dev/disk/` zu finden und werden vom System automatisch erstellt. Für `Label` gilt dies nur, sofern diese den Blockgeräten zuvor zugewiesen wurden.

1. **UUID**
   Er ist eine eindeutige Kennung auf Dateisystem-Ebene und in den Metadaten des Dateisystems gespeichert. Zum Auslesen muss der Dateisystemtyp bekannt und lesbar sein. Er ist unique (einzigartig), denn bereits beim Formatieren einer Partition wird ein neuer UUID erstellt.
   Ein UUID ist eine 128-Bit-Zahl. Jeder kann einen UUID erstellen und ihn verwenden. Die Wahrscheinlichkeit, dass ein UUID dupliziert wird, ist zwar nicht null, aber so gering, dass der Fall vernachlässigt werden kann. Alle Linux-Dateisysteme inklusive swap unterstützen UUID. Obwohl FAT- und NTFS-Dateisysteme UUID nicht unterstützen, werden sie in `/dev/disk/by-uuid` gelistet.

2. **PARTUUID**
   Er ist eine Kennung auf Partitionstabellen-Ebene die mit GTP eingeführt wurde. Er bleibt erhalten wenn die Partition umformatiert wird und ist damit nicht unique. Zum Beispiel scheitert das Mounten mittels eines fstab Eintrages auf Basis von PARTUUID, wenn die Partition mit einem anderen Dateisystem versehen wurde ohne die fstab anzupassen.

3. **Geräte-ID (ID)**
   Die ID wird aus den Metadaten des Gerätes (Hersteller, Anschlussart, Bauart, Speichervolumen usw.) erstellt und berücksichtigt weder die Partitionierung,

noch die Dateisysteme in den Partitionen. Sie ist als dauerhafter Bezeichner in der fstab ungeeignet.

4. **PATH**
   Er setzt sich aus der Bezeichnung des Controllers, der Geräteart und der Partitionsnummer zusammen. Wie bei der ID ist er als dauerhafter Bezeichner in der fstab ungeeignet.

5. **LABEL**
   Label sind von uns selbst vergebene, leicht wiedererkennbare Bezeichner. Sie sind nicht unique, deshalb muss sehr genau darauf geachtet werden Namensüberschneidungen zu vermeiden.

In der Grundeinstellung benutzt siduction aus oben genannten Gründen UUID in der `/etc/fstab`.

4.4.2 Label verwenden

Das Label eines Blockgerätes hat für uns Menschen den Vorteil leicht verständlich und gut wiedererkennbar zu sein. Praktisch jeder Typ von Dateisystem kann ein Label haben. Partitionen mit einem Label findet man im Verzeichnis `/dev/disk/by-label`:

```
$ ls -l /dev/disk/by-label
total 0
lrwxrwxrwx 1 root root 10 Oct 16 10:27 data -> ../../sdb2
lrwxrwxrwx 1 root root 10 Oct 16 10:27 home -> ../../sda6
lrwxrwxrwx 1 root root 10 Oct 16 10:27 root -> ../../sda1
lrwxrwxrwx 1 root root 10 Oct 16 10:27 swap -> ../../sda5
lrwxrwxrwx 1 root root 10 Oct 16 10:27 windows -> ../../sdb1
```

Die Bezeichnung eines Labels kann je nach Dateisystem mit folgenden Befehlen erzeugt bzw. geändert werden:

- **swap**
  `swaplabel -L <label> /dev/sdXx`
- **ext2/ext3/ext4**
  `e2label /dev/sdXx <label>` oder `tune2fs -L <label> /dev/sdXx`

- **jfs**
  `jfs_tune -L <label> /dev/sdXx`
- **xfs**
  `xfs_admin -L <label> /dev/sdXx`
- **ReiserFS**
  `reiserfstune -l <label> /dev/sdXx`
- **fat**
  `fatlabel /dev/sdXx <label>`
- **ntfs**
  `ntfslabel /dev/sdXx <label>`

Der Name des Labels einer NTFS- und FAT-Partition sollte nur aus Großbuchstaben, Ziffern und den für Dateinamen erlaubten Sonderzeichen von Windows™ bestehen.

Die Syntax in der `fstab` für das `<file system>` ist `LABEL=<label>`.

> Unbedingt zu beachten ist:
> Die Labels müssen eine singuläre Bezeichnung haben, um bei der Einbindung funktionieren zu können. Das gilt auch für externe Geräte (Festplatten, Sticks etc.), die via USB oder Firewire eingebunden werden.

4.5 Die fstab

Die Datei `/etc/fstab` wird während des Systemstarts ausgelesen um die gewünschten Partitionen einzuhängen. Hier ein Beispiel einer fstab.

```
# <file system>                          <mount point>  <type>  <↙
   options>  <dump><pass>
UUID=2e3a21ef-b98b-4d53-af62-cbf9666c1256 swap           swap↙
       defaults,noatime 0 2
UUID=1c257cff-1c96-4c4f-811f-46a87bcf6abb /              ext4↙
       defaults,noatime 0 1
UUID=35336532-0cc8-4613-9b1a-f31b12ea58c3 /home          ext4↙
       defaults,noatime 0 2
```

```
tmpfs                              /tmp           tmpfs    ↙
   defaults,noatime,mode=1777 0 0
UUID=e2164479-3f71-4216-a4d4-af3321750322 /mnt/TEST_root ext4↙
       noauto,noatime 0 0
LABEL=TEST_HOME                          /mnt/TEST_home ext4    ↙
   noauto,users,noatime 0 0
UUID=B248-1CCA                         /mnt/TEST_boot vfat    ↙
   noauto,users,rw,noatime 0 0
UUID=a7aeabe9-f09d-43b5-bb12-878b4c3d98c5 /mnt/TEST_res  ext4↙
       noauto,users,rw,noatime 0 0
```

Partitionen, die in der fstab aufgeführt sind, kann man mit ihrem “<file system>”-Bezeichner oder mit dem “<mount point>” einhängen.

```
$ mount UUID=a7aeabe9-f09d-43b5-bb12-878b4c3d98c5
    oder
$ mount /mnt/TEST_res
    oder
$ mount LABEL=TEST_HOME
```

4.5.1 Anpassung der fstab

Um neu erstellte Partitionen nutzen zu können (nehmen wir `sda5` und `sdb7` als Beispiele), die nicht in der fstab erscheinen oder sich nicht mit den zuvor genannten Befehlen mounten lassen, tippt man als **user** folgenden Befehl in die Konsole:

```
ls -l /dev/disk/by-uuid
```

Er wird etwas Ähnliches wie dies hier ausgeben:

```
lrwxrwxrwx 1 root root 10 Mai 29 17:51 1c257cff-1c96-4c4f-811↙
   f-46a87bcf6abb -> ../../sda2
lrwxrwxrwx 1 root root 10 Mai 29 17:51 2e3a21ef-b98b-4d53-↙
   af62-cbf9666c1256 -> ../../sda1
lrwxrwxrwx 1 root root 10 Mai 29 17:51 2ef32215-d545-4e12-↙
   bc00-d0099a218970 -> ../../sda5
lrwxrwxrwx 1 root root 10 Mai 29 17:51 35336532-0cc8-4613-9↙
   b1a-f31b12ea58c3 -> ../../sda4
```

```
lrwxrwxrwx 1 root root 10 Mai 29 17:51 4c4b9246-2904-40d1-⇙
   addc-724fc90a2b6a -> ../../sdb3
lrwxrwxrwx 1 root root 10 Mai 29 17:51 a7aeabe9-f09d-43b5-⇙
   bb12-878b4c3d98c5 -> ../../sdb7
lrwxrwxrwx 1 root root 10 Mai 29 17:51 B248-1CCA -> ../../⇙
   sdb1
lrwxrwxrwx 1 root root 10 Mai 29 17:51 d5b01bbc-700c-43ce-⇙
   a382-1ba95a59de78 -> ../../sdb6
lrwxrwxrwx 1 root root 10 Mai 29 17:51 e2164479-3f71-4216-⇙
   a4d4-af3321750322 -> ../../sdb5
lrwxrwxrwx 1 root root 10 Mai 29 17:51 f5ed412d-7b7b-41c1-80⇙
   ce-53337c82405b -> ../../sdb2
```

In diesem Beispiel ist
`2ef32215-d545-4e12-bc00-d0099a218970` der fehlende Eintrag für `sda5` und
`a7aeabe9-f09d-43b5-bb12-878b4c3d98c5` der fehlende Eintrag für `sdb7`.

Der nächste Schritt ist, die UUID/Partitionen in die `/etc/fstab` einzutragen. Um sie zu dieser hinzuzufügen, benutzt man einen Texteditor (wie mcedit, kate, kwrite oder gedit) mit Rootrechten; in diesem Beispiel sähe der Eintrag so aus:

```
# <file system>                            <mount point>     ⇙
   <type>  <options> <dump><pass>
UUID=2ef32215-d545-4e12-bc00-d0099a218970  /media/disk1part5 ⇙
   ext4 auto,users,exec 0 2
UUID=a7aeabe9-f09d-43b5-bb12-878b4c3d98c5  /media/disk2part7 ⇙
   ext4 auto,users,exec 0 2
```

4.5.2 Erstellung neuer Einhängepunkte

Anmerkung: Ein Einhängepunkt, der in fstab festgelegt wird, muss einem existierenden Verzeichnis zugeordnet sein. Diese Verzeichnisse werden während der Live-Session von siduction unterhalb von `/media` angelegt und besitzen das Benennungsschema `diskXpartX` .

Wenn nun die Partitionierungstabelle nach der Installation verändert und fstab angepasst wurde (zum Beispiel wurden zwei neue Partitionen angelegt), existiert noch kein Einhängepunkt. Er muss manuell angelegt werden.

Beispiel

Als erstes werden wir zu **root** und ermitteln die bestehenden Einhängepunkte. Die Ausgabe zeigt zum Beispiel:

```
cd /media
ls
disk1part1 disk1part3 disk2part1
```

Im Verzeichnis `/media` werden nun die Einhängepunkte der neuen Partitionen angelegt:

```
mkdir disk1part5
mkdir disk2part7
```

So können die neuen Partitionen sofort genutzt oder getestet werden:

```
mount /media/disk1part5
mount /media/disk2part7
```

Nach einem Neustart des Computers werden die neuen Dateisysteme automatisch eingebunden wenn in der fstab unter “<options>” `auto` oder `defaults` eingetragen ist. Siehe auch:

```
man mount
```

Natürlich muss man sich nicht an das Namensschema *“diskXpartX”* halten. Einhängepunkte (mountpoints) und die dazugehörigen Bezeichner in der fstab können sinnvoll mit z.B. *“data”* oder *“music”* benannt werden.

Zuletzt bearbeitet: 2021-11-28

4.6 Partitionieren mit GParted

Partitionen zu erstellen oder zu bearbeiten ist keine alltägliche Aufgabe. Daher ist es eine gute Idee, folgende Anleitung einmal gelesen zu haben, um mit dem Konzept eines Partitionsmanagers vertraut zu werden.

4.6.1 Wichtige Hinweise

- Zuerst immer ein Daten-Backup anlegen!
- Bezüglich der Benennung von Speichergeräten das Kapitel zu UUID, Partitionsbezeichnung und fstab zu Rate ziehen, da siduction in der Grundeinstellung Benennung nach UUID verwendet.
- Größenänderungen bei **NTFS-Partitionen** erfordern nach der Ausführung einen sofortigen Reboot, vorher dürfen keine weiteren Änderungen an Partitionen durchgeführt werden. Dies führte unweigerlich zu Fehlern. Bitte lese hier weiter.
- Eine Partition benötigt ein Dateisystem. Linux kann auf und mit verschiedenen Dateisystemen arbeiten.
  Für normalen Gebrauch empfehlen wir das Dateisystem **ext4**.
  NTFS sollte man verwenden, wenn die Partition auch von einer Windows-Installation benutzt werden soll. Siduction kann mit dem automatisch installierten `ntfs-3g` lesend und schreibend auf die Daten zugreifen.
- Die gesamte GParted-Dokumentation findet sich in vielen Sprachen auf der GParted-Homepage.

4.6.2 GParted verwenden

Der Programmstarter für GParted befindet sich in

- **KDE, LXQt, XFCE**
  im Anwendungsmenü - System - GParted
- **Gnome**
  in Anwendungen - Gparted

Nach dem Klick auf den Starter öffnet sich ein Dialog zur Abfrage des Root-Passwortes.

Wenn GParted startet, öffnet sich das Programmfenster und die vorhandenen Laufwerke werden ausgelesen.

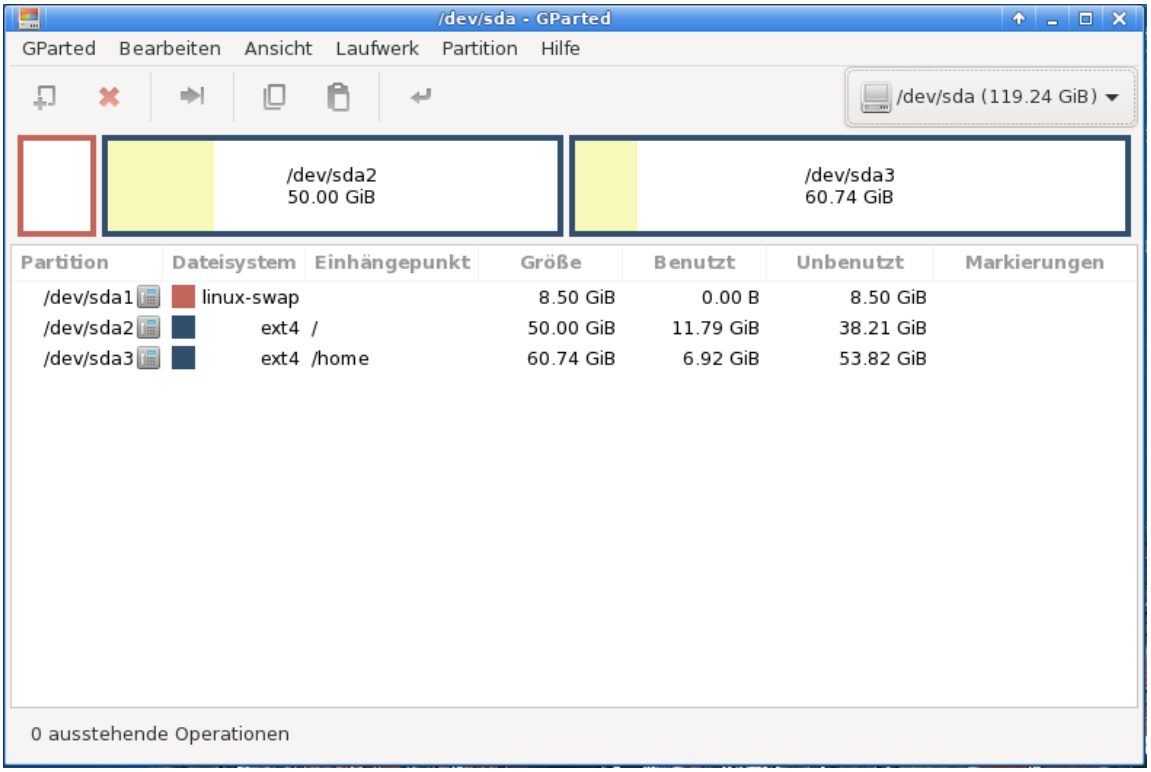


Abbildung 14: GParted Startfenster

Der erste Menüpunkt `GParted` öffnet eine Drop-Down-Liste, zum erneuten Einlesen der Laufwerke, zur Auswahl eines Laufwerkes oder zum Beenden des Programms.

- **Bearbeiten**

  `Bearbeiten` ist der 2. Menüpunkt von links. Er zeigt drei ausgegraute Optionen, die sehr wichtig sind und weiter unten erläutert werden.

  – letzte Operationen rückgängig machen (“Undo last operations”),

  – alle Operationen löschen (“clear all operations”) und

  – alle Operationen ausführen (“apply all operations”).

- **Ansicht**

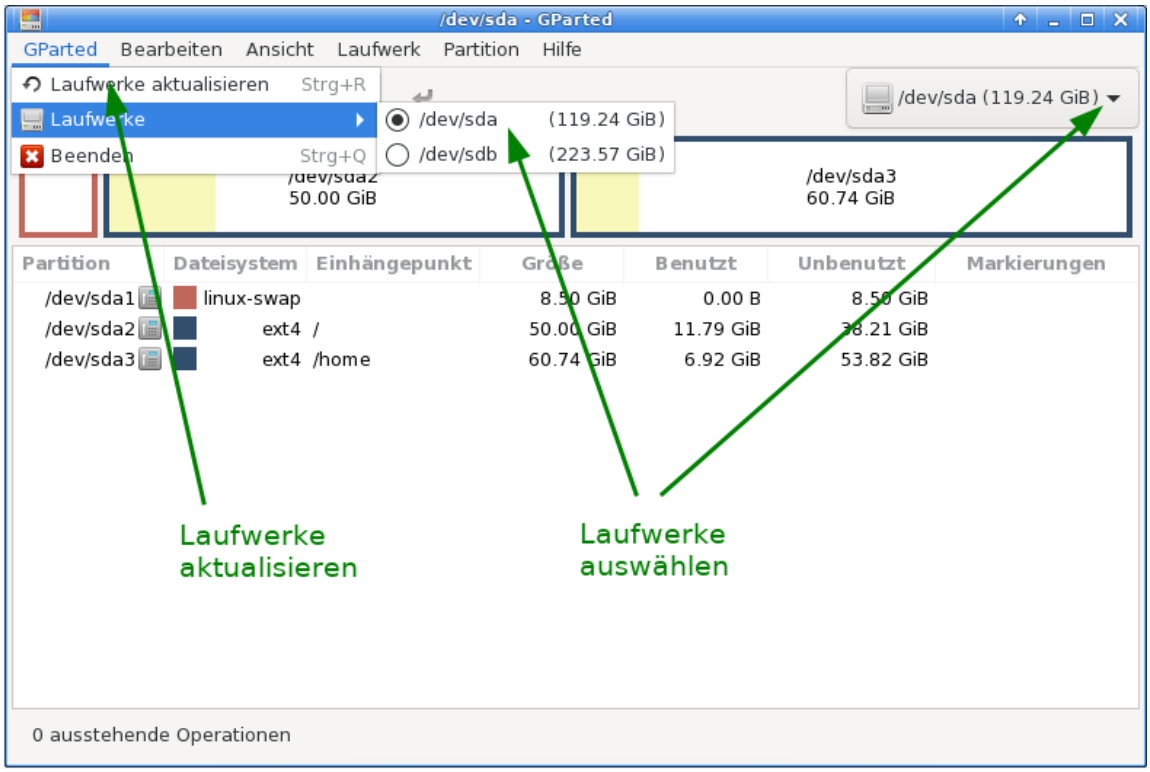


Abbildung 15: GParted Geräteübersicht

Der nächste Menüpunkt `Ansicht` bietet die Anzeigeoptionen `Laufwerksinformationen` und `Anstehende Operationen`.

- Laufwerksinformationen (“Device Information”)
  Im linken Rahmen stehen Details der Laufwerke wie Modell, Größe usw., die wichtig sind, wenn mehrere Datenträger im System vorhanden sind. Damit kann man kontrollieren, ob der richtige Datenträger zur Formatierung gewählt wurde.

- Anstehende Operationen (“Pending Operations”)
  In einem unten sich öffnenden Rahmen werden die ausstehenden Operationen angezeigt. Diese Information ist sehr nützlich, um einen Überblick darüber zu haben, welche Operationen durchgeführt werden sollen. Der Rahmen öffnet sich auch automatisch, sobald für ein Laufwerk eine Operationen angefordert wird.

Die beiden Bereiche sind grün markiert.

- **Laufwerk**

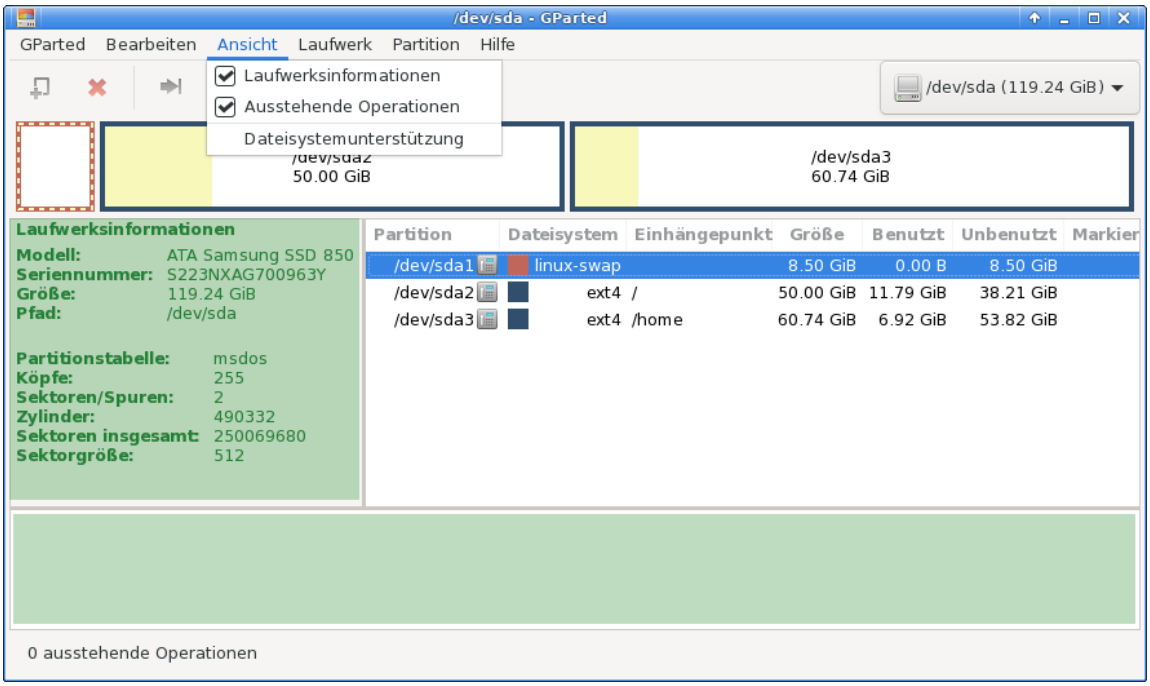


Abbildung 16: GParted Festplatteninformation

Hinter dem Menüpunkt `Partitionstabelle erstellen` verbergen sich eigentlich zwei Optionen

1. Eine neue (leere) Partitionstabelle des gleichen Typ erstellen, und damit auf dem schnellsten Weg alle alten Partitionen und Daten zu entfernen.

2. Einen Wechsel des Typ der Partitionstabelle vorzunehmen. Sinnvoller Weise von msdos-MBR zu gpt-UEFI oder umgekehrt. Auch hierbei gehen alle Daten verloren.
   Im Jahr 2009 wurde das UEFI mit GPT eingeführt, hat sich seitdem nach und nach verbreitet, und wird den MBR ersetzen. Zwar unterstützen moderne UEFI-Mainboard MBR, die Vorteile von GPT gehen dabei jedoch verloren. Weitere Informationen zu UEFI und GPT liefert die Handbuchseite Partitionieren mit gdisk.

Die Auswahl *“Datenrettung versuchen”* bietet bei Erfolg die Chance trotz einer defekten Partitionstabelle doch noch an die Daten zu gelangen.

- **Partition**

Der Menüpunkt `Partition` ist von größter Wichtigkeit. Für die unten ausgewählte Partition zeigt das Menü alle zur Verfügung stehenden Operationen

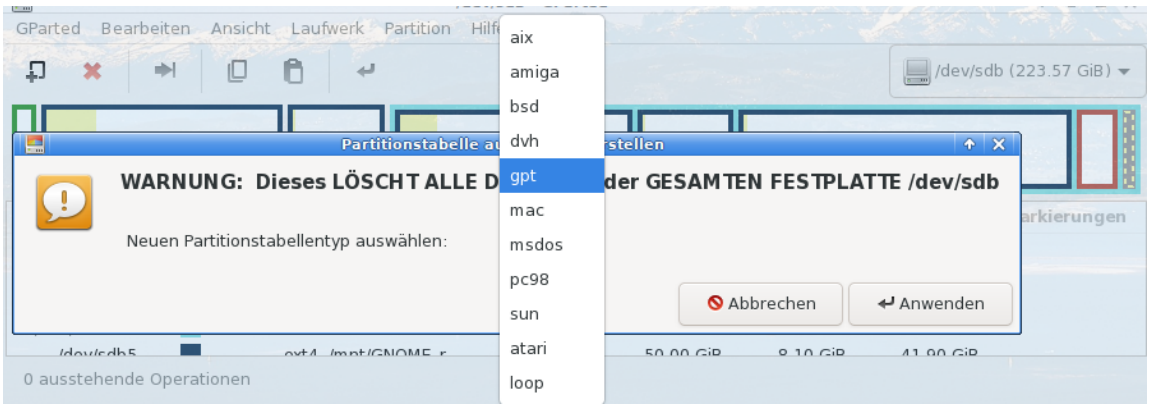


Abbildung 17: GParted Partitionstabelle

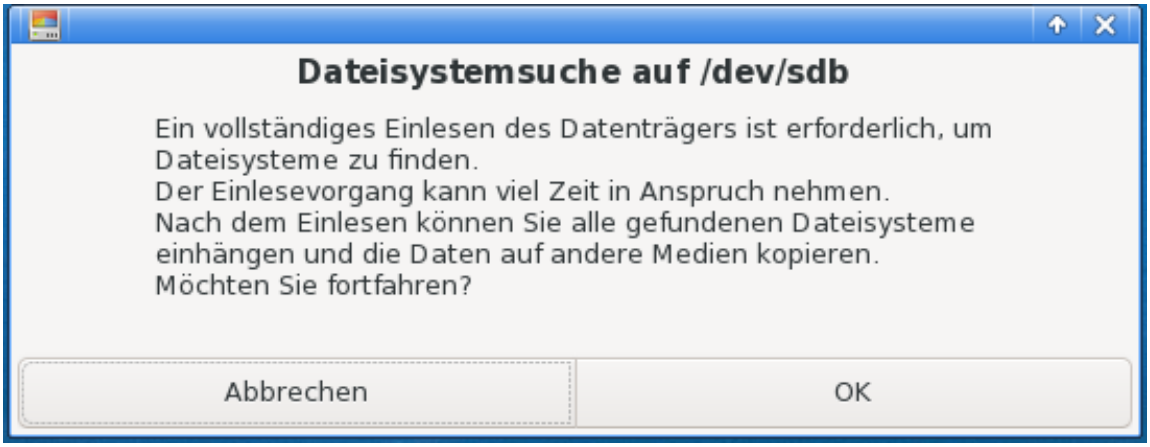


Abbildung 18: GParted Datenrettung

abhängig davon an, ob die Partition eingehangen oder nicht eingehangen ist. Beachten sollte man, dass einige der Unterpunkte auch kritische bzw. gefährliche Aktionen durchführen können.

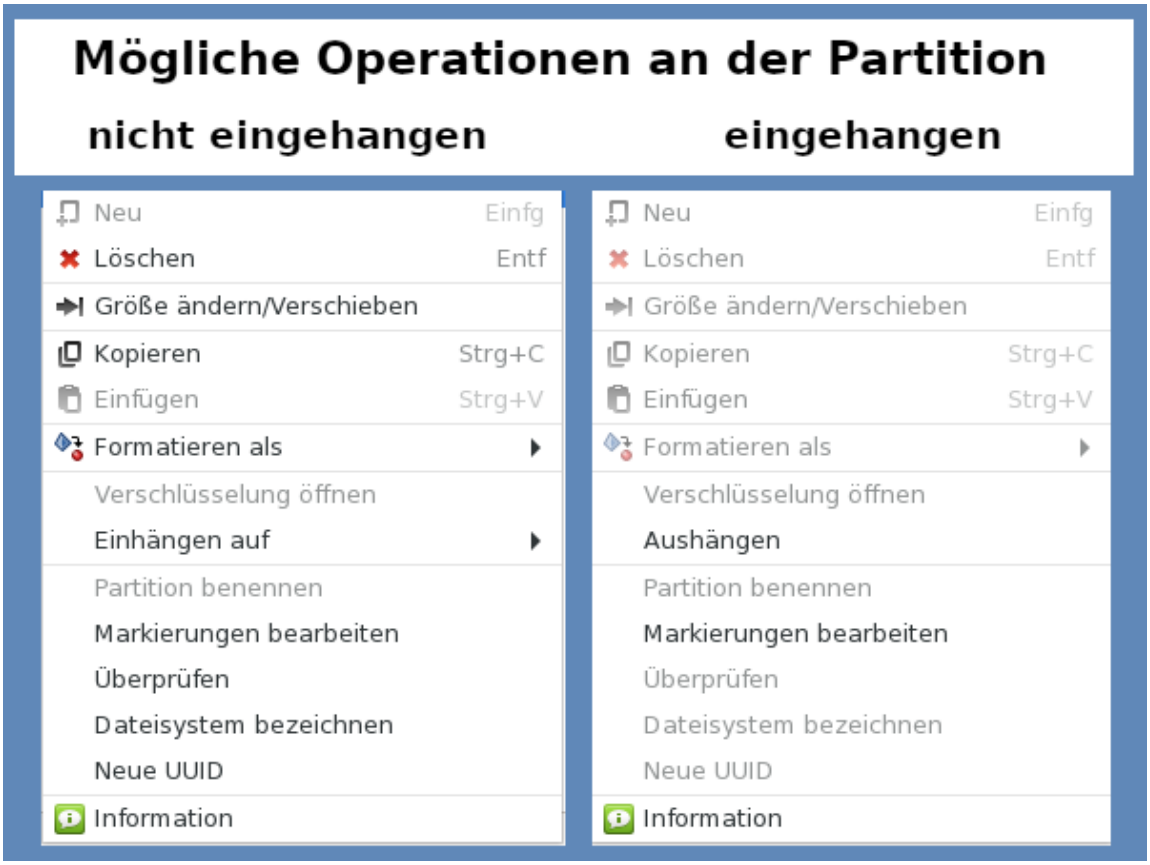


Abbildung 19: GParted Datenrettung

- **Eine neue Partition erstellen**

  In der Toolbar erlaubt der Knopf **Neu** das Erstellen einer neuen Partition, wenn zuvor ein nicht zugeordneter Bereich gewählt wurde. Im nächsten Fenster erfolgt die Festlegung der Größe für eine primäre, erweiterte oder logische Partition und die Festlegung des Dateisystems.

- **Größe ändern/verschieben**

  Die Partition kann mit der Maus verkleinert, vergrößert und verschoben werden. Alternativ trägt man die neuen Werte in die dafür vorgesehenen Felder ein.

- **Falls ein Fehler gemacht wurde**

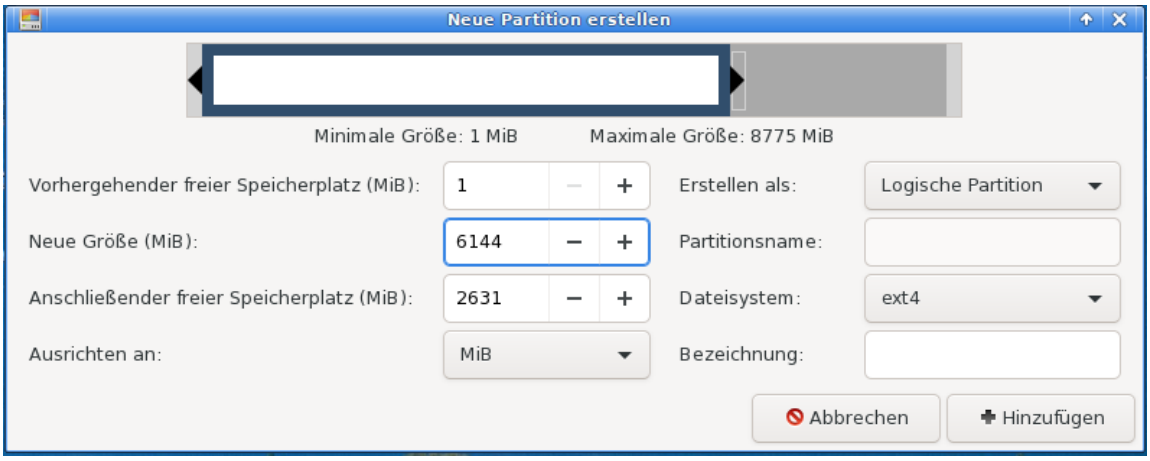


Abbildung 20: GParted Neue Partition

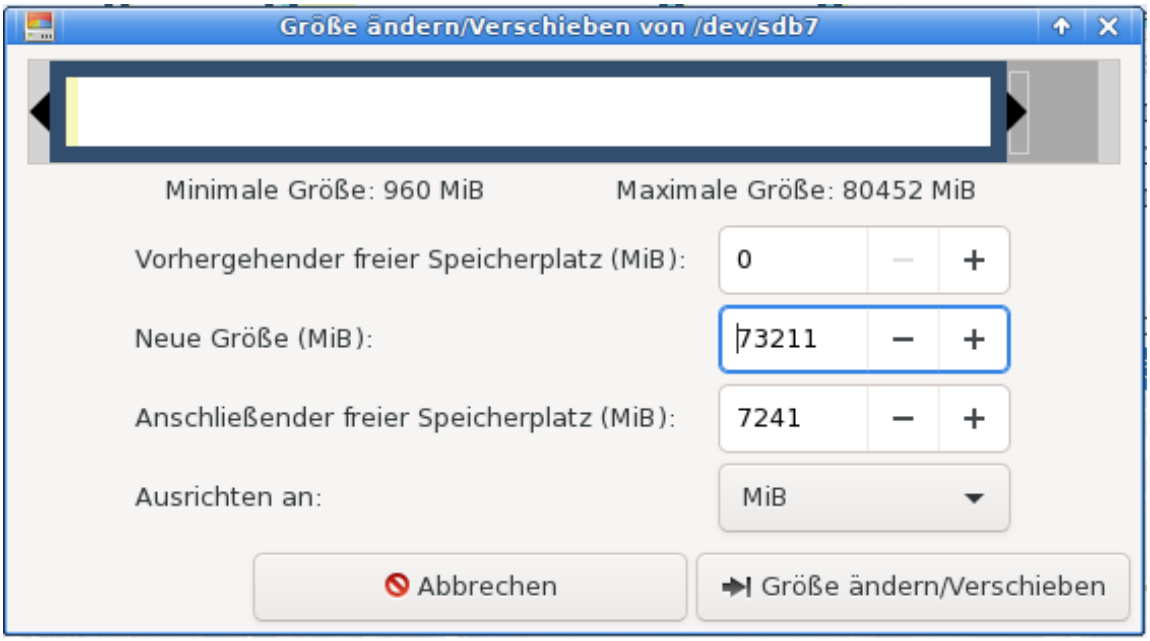


Abbildung 21: GParted Größenänderung

Im Menü `Bearbeiten` besteht die Möglichkeit `Letzte Operation rückgängig machen` oder `Alle Operationen löschen`. Der Bereich ist grün markiert.

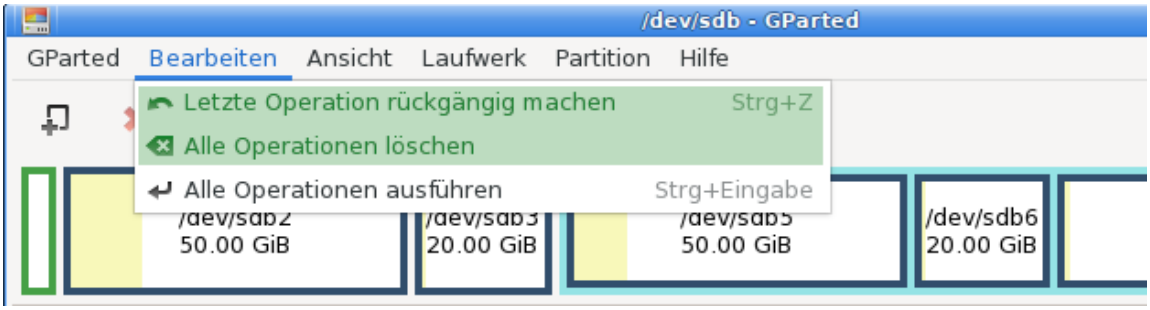


Abbildung 22: GParted rückgängig machen

- **Anwenden**

  Bis jetzt wurden noch keine Änderungen auf den Laufwerken vorgenommen. Wenn man sicher ist, dass alle vorgesehenen Änderungen richtig sind, wählt man im Menü `Bearbeiten` den Punkt `Alle Operationen ausführen`. Darauf erscheint der folgende Dialog, der zu bestätigen ist.

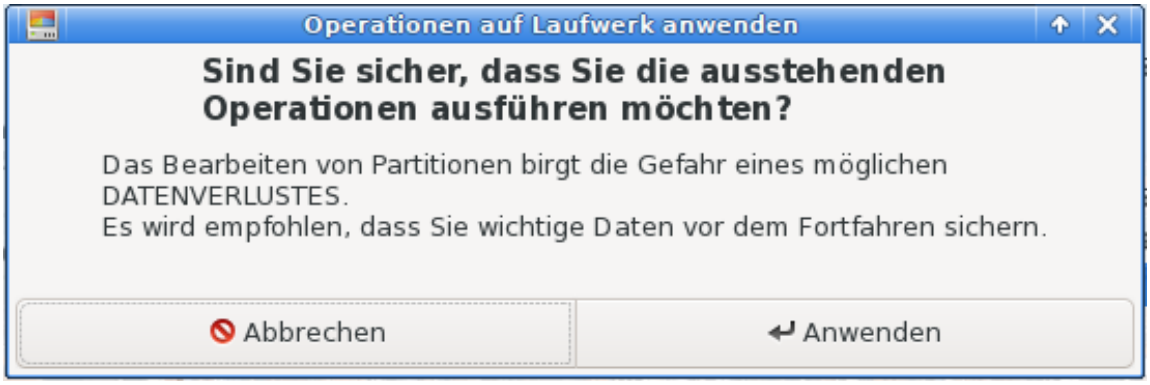


Abbildung 23: GParted Ausführen und speichern

Die Dauer der Operation hängt von der Größe der gewählten Partition ab.

4.6.3 fstab anpassen

Nachdem die Änderungen auf die Laufwerke geschrieben wurden, muss die Datei `/etc/fstab` überprüft und ggf. angepasst werden.

Siehe dazu die Handbuchseite Anpassung der fstab.
In einem root-Terminal geben wir die Befehle **`cat /etc/fstab`** und **`blkid`** ein und vergleichen die UUIDs.

```
root@pc1:/# cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for ↙
   a device; this may
# be used with UUID= as a more robust way to name devices ↙
   that works even if
# disks are added and removed. See fstab(5).
#
# <file system>          <mount point>  <type>  <options> <↙
   dump><pass>
UUID=2e3a21ef-b98b-4d53-af62-cbf9666c1256 swap           swap↙
       defaults,noatime 0 2
UUID=1c257cff-1c96-4c4f-811f-46a87bcf6abb /              ext4↙
       defaults,noatime 0 1
UUID=35336532-0cc8-4613-9b1a-f31b12ea58c3 /home          ext4↙
       defaults,noatime 0 2
tmpfs                           /tmp           tmpfs    ↙
   defaults,noatime,mode=1777 0 0
UUID=f5ed412d-7b7b-41c1-80ce-53337c82405b /mnt/Foto      ext4↙
       defaults,noatime 0 0
UUID=4c4b9246-2904-40d1-addc-724fc90a2b6a /mnt/Backup    ext4↙
       noauto,users,noatime 0 0
UUID=a7aeabe9-f09d-43b5-bb12-878b4c3d98c5 /mnt/TEST_res  ext4↙
       noauto,users,rw,noatime 0 0
```

```
root@pc1:/# blkid
/dev/sda1: UUID="2e3a21ef-b98b-4d53-af62-cbf9666c1256" TYPE="↙
   swap" PARTUUID="000403b7-01"
/dev/sda2: UUID="1c257cff-1c96-4c4f-811f-46a87bcf6abb" ↙
   BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="000403b7-02"
/dev/sda3: UUID="35336532-0cc8-4613-9b1a-f31b12ea58c3" ↙
   BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="000403b7-03"
/dev/sdb1: UUID="f5ed412d-7b7b-41c1-80ce-53337c82405b" ↙
   BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="2853e345-01"
```

```
/dev/sdb2: UUID="4c4b9246-2904-40d1-addc-724fc90a2b6a" ⇙
   BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="2853e345-02"
/dev/sdb5: UUID="e2164479-3f71-4216-a4d4-af3321750322" ⇙
   BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="2853e345-05"
/dev/sdb6: UUID="2ef32215-d545-4e12-bc00-d0099a218970" ⇙
   BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="2853e345-06"
```

Wir können erkennen, dass die in der fstab als letzter Eintrag enthaltene, nach *"/mnt/TEST_res"* eingehängte Partition in der blkid-Liste nicht mehr enthalten ist. Dafür haben wir zwei neue Partitionen. Bei diesem Beispiel würde der PC einen Reboot zwar durchführen, jedoch *"/mnt/TEST_res"* und die zwei neuen Partitionen nicht automatisch einhängen können. Der Bootvorgang würde sich erheblich verzögern.

Wenn die UUID für die Partitionen von `/` (root), `/home` und `swap` nicht mit dem jeweiligen Eintrag in der `/etc/fstab` übereinstimmen, müssen die Einträge zwingend angepasst werden, sonst fährt das System nach einem Reboot nicht mehr hoch.

4.6.4 NTFS-Partitionsgrößen mit GParted ändern

Größenänderungen bei NTFS-Partitionen erfordern nach der Ausführung einen sofortigen Reboot, vorher dürfen keine weiteren Änderungen an Partitionen durchgeführt werden. Dies führte unweigerlich zu Fehlern.

- Nach dem Neustart von Windows und dem Windows-Logo erscheint ein Fenster von **`checkdisk`**, das besagt, dass **`C:\`** auf Fehler überprüft wird.
- Diesen AUTOCHECK bitte zu Ende laufen lassen: Windows muss das Filesystem nach einer Größenänderung überprüfen.
- Nach der Überprüfung wird der Rechner automatisch das zweite Mal neu gestartet. Dies gewährleistet, dass das System problemlos laufen kann.
- Nach dem Neustart wird Windows ordnungsgemäß funktionieren. Man muss jedoch das System fertig starten lassen und auf das Anmeldefenster warten!

Zuletzt bearbeitet: 2021-11-28

4.7 Partitionieren mit gdisk

Warum gdisk (GPT fdisk) verwenden?

gdisk leitet sich von **G**lobally Unique Identifier **P**artition **T**able (GPT) ab und ist eine Anwendung, um Datenträger von jeder Größe zu partitionieren. gdisk wird unbedingt benötigt für Datenträger, die größer als 2TB sind.
gdisk sorgt dafür, dass Partitionen für SSDs eingerichtet sind (bzw. für Speicher, die keine 512 Byte großen Sektoren besitzen).

Ein entscheidender Vorteil von GPT ist, dass man nicht mehr auf die dem MBR inhärenten primären, erweiterten oder logischen Partitionen angewiesen ist. GPT kann eine beinahe unbegrenzte Anzahl von Partitionen unterstützen und ist nur durch den für Partitionseinträge reservierten Speicherplatz des GPT-Datenträgers eingeschränkt. Zu beachten ist, dass die Anwendung gdisk standardmäßig für 128 Partitionen eingestellt ist.

Falls GPT auf kleinen USB/SSD-Datenträgern eingesetzt wird (zum Beispiel auf einem USB-Stick mit 8GB), könnte sich dies kontraproduktiv auswirken, wenn Daten zwischen verschiedenen Computern oder Betriebssystemen ausgetauscht werden sollen.
Für diesen Zweck, und sofern ältere Hardware zum Einsatz kommt, verwenden wir besser `fdisk`, das Partitionstabellen auf Basis des MBR erstellt. Siehe die Handbuchseite Partitionieren mit Cfdisk.

Wichtige Anmerkungen

- Die Begriffe UEFI und EFI sind austauschbar und bezeichnen das gleiche Konzept - **U**nified **E**xtensible **F**irmware **I**nterface (englisch für Vereinheitlichte erweiterbare Firmware-Schnittstelle).
  Siehe Wikipedia UEFI.
  Die GPT ist ein Teil des UEFI Standards.
- GPT-Datenträger verwenden
  - GPT-Datenträger können unter Linux auf Computern mit 32 bit und 64 bit eingesetzt werden.
  - Einige Betriebssysteme unterstützen keine GPT-Datenträger.
    Dazu zählen alle MS Betriebssysteme vor Windows Vista SP1.
    Ziehe bitte die Dokumentation des jeweiligen Systems zu Rate.

- Booten von GPT-Datenträgern
  - Dual- und Triple-Boot von GPT-Datenträgern mit Linux, BSD und Apple ist mit dem EFI-Modus mit 64 bit unterstützt.
  - Dual-Boot von GPT-Datenträgern mit Linux und MS Windows ist ab Windows Vista SP1 möglich. Voraussetzung ist dabei für Windows die 64 bit Version.
- Graphische Partitionierungsprogramme für GPT
  Neben dem Befehlszeilenprogramm `gdisk` unterstützen graphische Anwendungen wie `gparted` und `partitionmanager` GPT-Datenträger. Trotzdem empfehlen wir gdisk, um unerwünschten Anomalien vorzubeugen. Gparted sowie KDEs Partition Manager (und andere) sind dennoch großartige Hilfsmittel besonders um die Partitionierung zu visualisieren.

Grundlegende Lektüre:

- man gdisk
- GPT fdisk Tutorial by Roderick W. Smith (Englisch)
- Wikipedia UEFI-Unterstützung der Betriebssysteme
- Wikipedia GUID-Partitionstabelle (Deutsch)

4.7.1 Partitionierung einer Festplatte

Daten zuvor sichern!
Bei Verwendung jedweder Partitionierungssoftware droht Datenverlust. Daten, die erhalten bleiben sollen, immer zuvor auf einem anderen Datenträger sichern.

In dem folgenden Beispiel werden wir eine 150GB Festplatte so formatieren, dass anschließend zwei Linux Systeme als Dualboot installierbar sein werden. Damit die Vorteile des UEFI zum tragen kommen, benötigen wir in der GPT eine *"EFI-System"*-Partition und für die zweite Stufe des GRUB-Bootloader eine *"BIOS-boot"*-Partition.
Wir zeigen die notwendigen Arbeitsschritte mit dem Partitionierungsprogramm `cgdisk`, das GPT mit UEFI unterstützt.

cgdisk ist die Curses-basierte Programmvariante von gdisk. Sie bietet eine benutzerfreundliche Bedienoberfläche innerhalb des Terminals.
Die Navigation erfolgt mittels der Pfeiltasten.

- Für die Partitionen **`auf`** und **`ab`**
- Für die Aktionsauswahl **`rechts`** und **`links`**.
- Mit **`Enter`** wird die Auswahl bzw. Eingabe bestätigt.

4.7.2 cgdisk verwenden

Der Startbefehl in einem root-Terminal lautet: **`cgdisk /dev/sdX`**.

cgdisk startet mit einer Warnmeldung, wenn keine GPT gefunden wird.

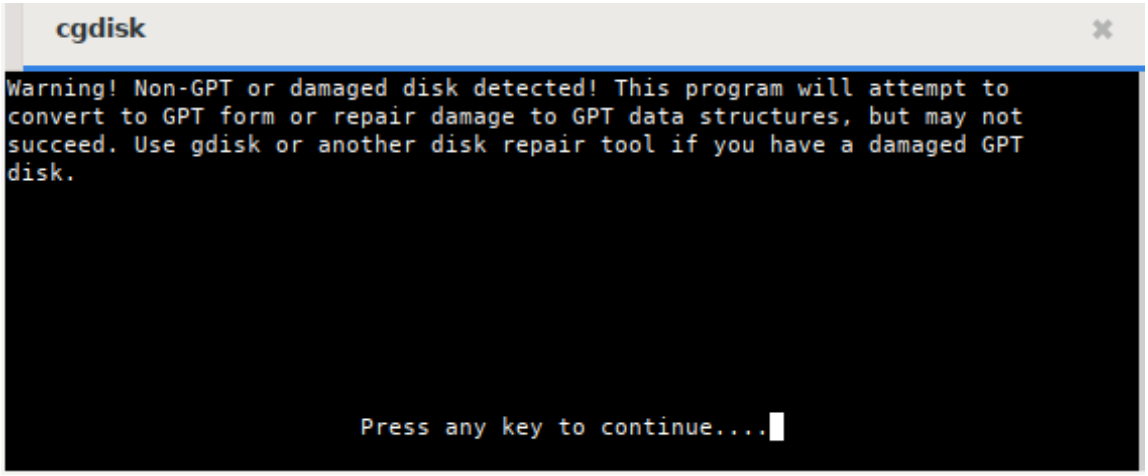


Abbildung 24: Warnmeldung

Wir benötigen für die beiden Betriebssysteme insgesamt sechs Partitionen: Zwei ROOT-, eine gemeinsame DATEN- sowie eine SWAP-Partition für den Auslagerungsspeicher. Zusätzlich die bereits oben erwähnte *“EFI-System”*-Partition (maximal 100MB) und die *“BIOS-boot”*-Partition (1MB).
Wir empfehlen, das `/home`-Verzeichnis auf der ROOT-Partition zu belassen. Das Verzeichnis `/home` sollte der Ort sein, an dem die individuellen Konfigurationen abgelegt werden, und nur diese. Für alle weiteren privaten Daten sollte eine eigene Datenpartition angelegt werden. Die Vorteile für die Datenstabilität, Datensicherung und auch im Falle einer Datenrettung sind nahezu unermesslich.

Das Startbild

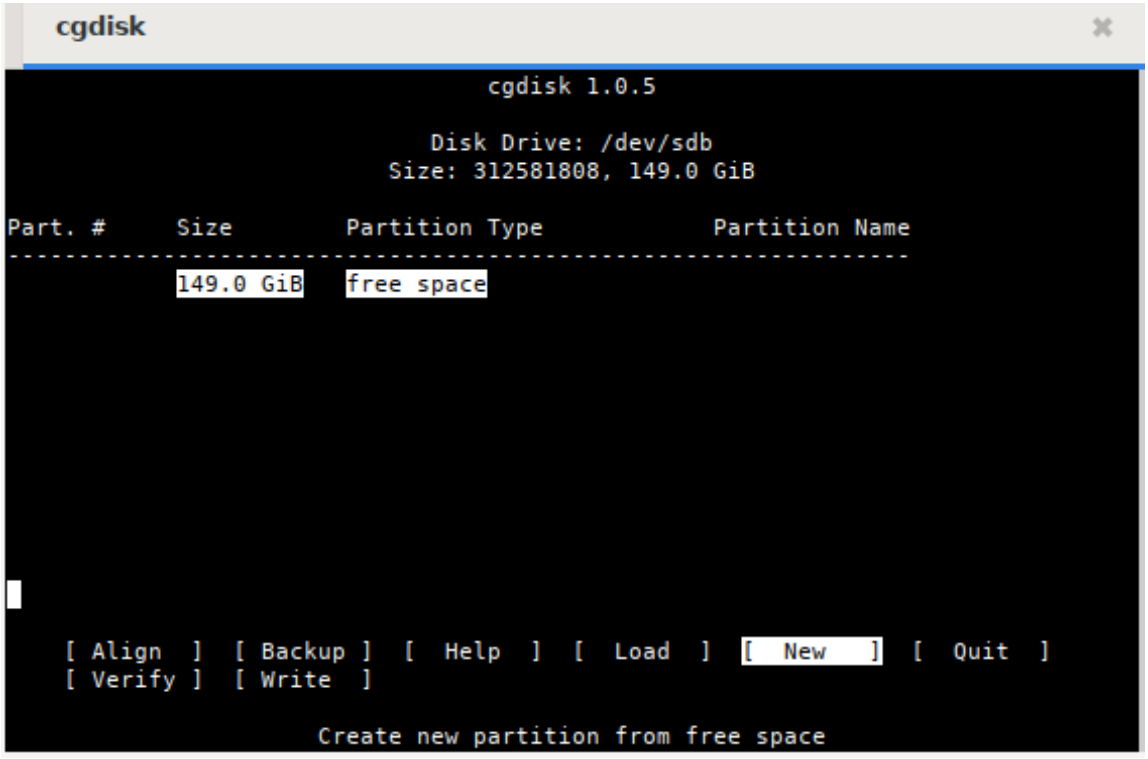


Abbildung 25: Startbild

4.7.2.1 Partition erstellen Wir wählen **New** und bestätigen mit **Enter**. Mit einem zweiten **Enter** übernehmen wir den voreingestellten ersten Sektor für die neue Partition. Dann geben wir die gewünschte Größe von `100M` für die *"EFI-System"*-Partition ein und bestätigen die Eingabe.

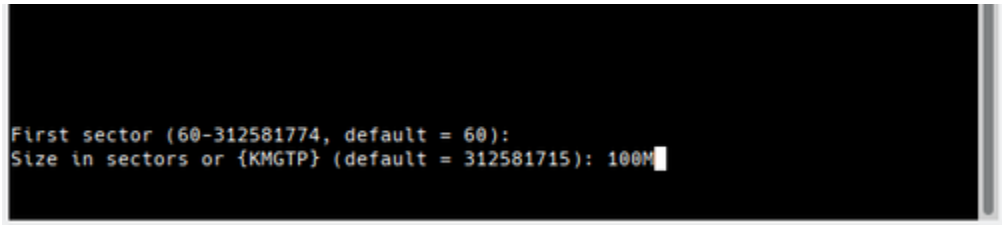


Abbildung 26: Neue Partition

Nun wird von uns die Eingabe des Type-Code für die Partition erwartet.

Nach Eingabe von **L** erscheint eine lange Liste mit den Codes und ihrer Verwendung. Die integrierte Suchfunktion vereinfacht die Auswahl. Für uns sind folgende Codes notwendig:
ef00 für EFI-System
ef02 für BIOS-boot

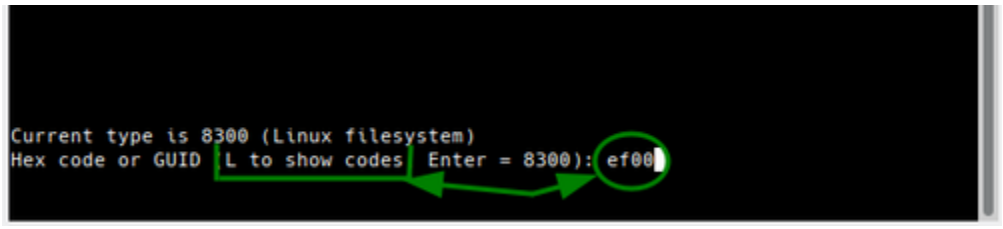


Abbildung 27: Type-Code

8200 für Swap
8304 für Linux Root
8300 für Linux Daten

Wir tragen also `ef00` ein und bestätigen. Anschließend dürfen wir optional noch einen Namen (Label) vergeben, was im Beispiel getan wurde, und die Eingabe wieder bestätigen. Mit den Partitionen für BIOS-boot, Linux-root und Swap wird nach gleichem Muster verfahren. Das nächst Bild zeigt das Ergebnis unserer Bemühungen. Wie wir sehen ist noch reichlich Platz für ein zweites System und vor allem für eine gemeinsam genutzte Daten-Partition vorhanden.

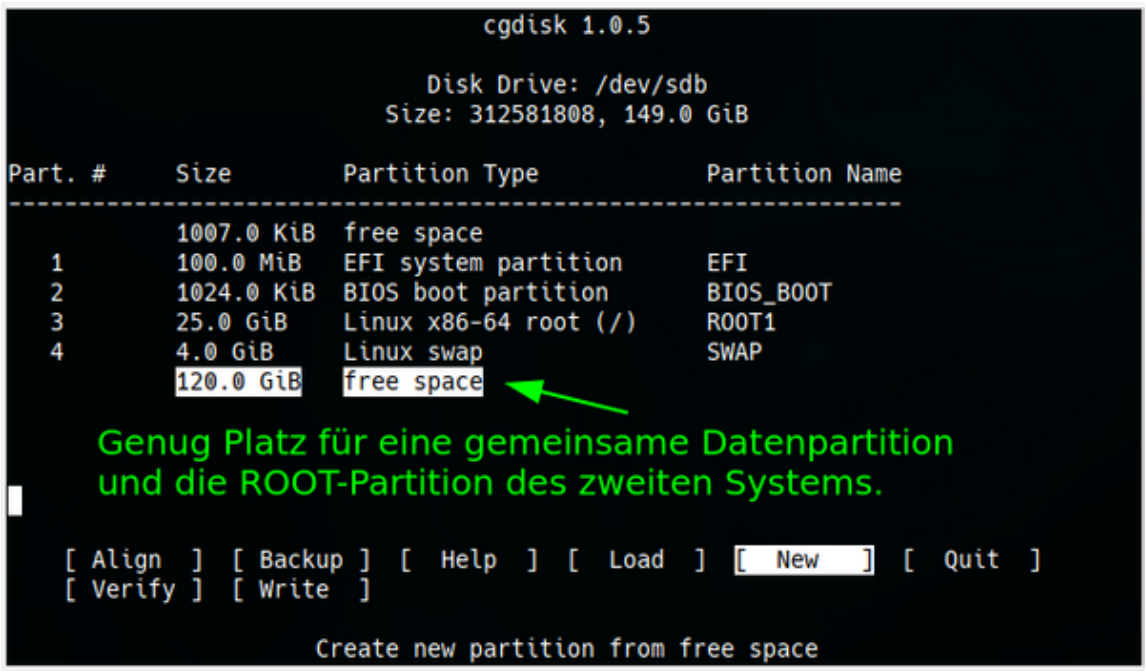


Abbildung 28: Erster Teil

Nachdem die zwei Partitionen erstellt wurden, sehen wir die Aufteilung der gesamten Festplatte im nächste Bild.

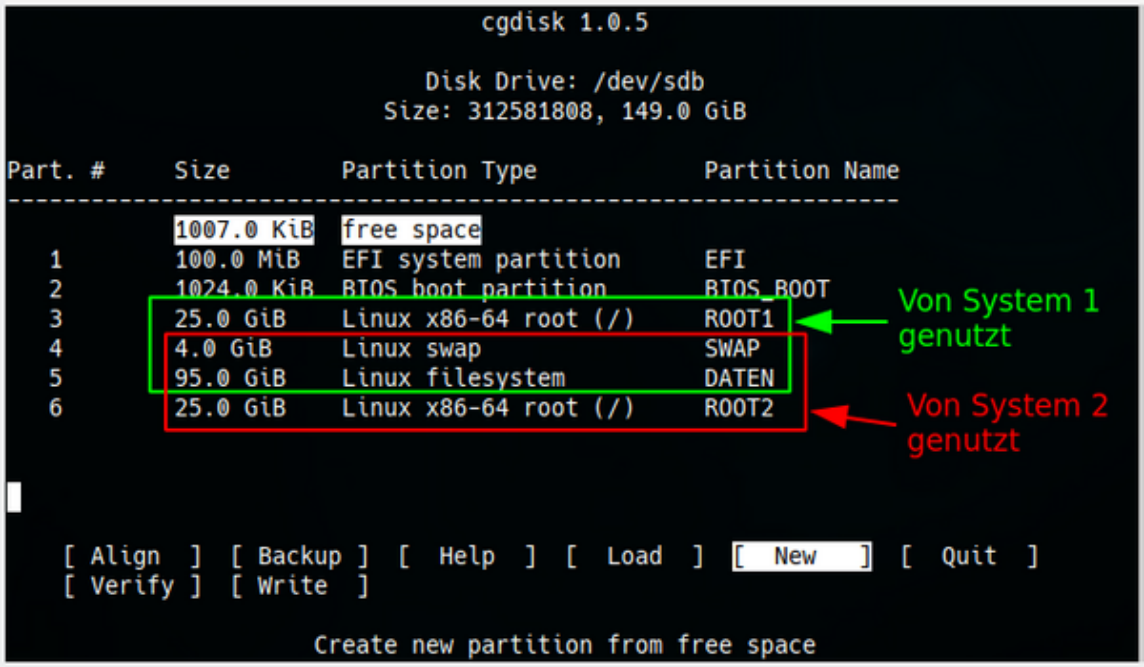


Abbildung 29: Gemeinsame Verwendung

Die Partitionen, die die beiden Systeme später im Betrieb verwenden, sind farblich gekennzeichnet.
An Anfang und Ende befinden sich noch kleine, freie Bereiche. Sie entstehen durch die Ausrichtung der Partition an die Blockgrenzen der Festplatte und können auch zwischen den Partitionen auftauchen. Mit **`Align`** kann der Wert für die Anzahl der Sektoren geändert werden. Für SSD und M2-Disk sind es in der Regel 2048 Sektoren und für alte Festplatten 512 Sektoren. gdisk liest die Metadaten der Festplatten aus und stellt den Wert für die Sektoren danach ein. Deshalb ist in der Regel keine Änderung notwendig.

Zusätzliche, detailierte Informationen zu den Partitionen lassen sich einsehen, wenn der Befehl **`Info`** benutzt wird.

Mit *Verify* wird die Partitionierung überprüft und eventuelle Fehler werden angezeigt.

Hier ist alles in Ordnung.
Sollten Fehler gemeldet werden, markieren wir die Partition und benutzen den Befehl **`Info`**, und entscheiden ob die Partition gelöscht und neu angelegt werden muss und ob dabei z. B. die Größe zu ändern ist. Wenn mit diesen Mitteln eine

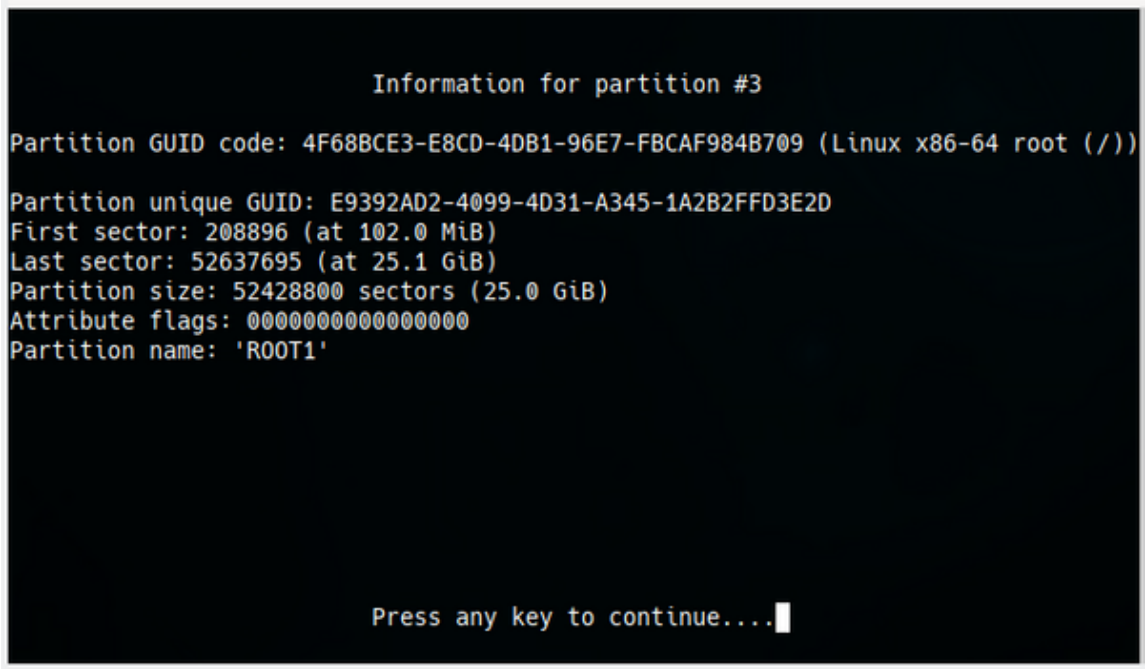


Abbildung 30: Partition Details

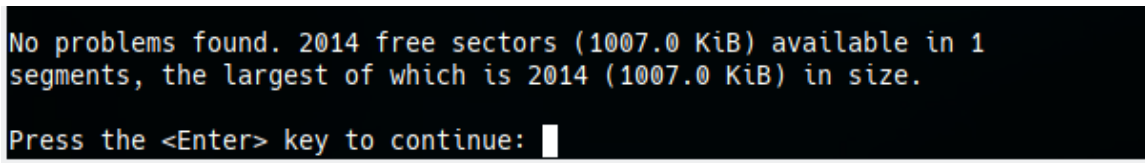


Abbildung 31: Partition Verify

Reparatur nicht möglich ist, stehen routinierten Usern die Erweiterten Befehle von gdisk zur Verfügung.

4.7.2.2 Partition löschen Um eine Partition zu löschen, markieren wir diese und benutzen den Befehl **`Delete`**.

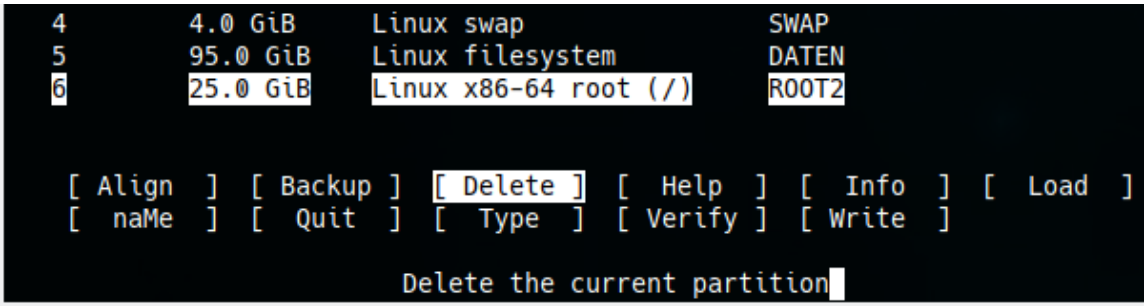


Abbildung 32: Partition löschen

Bei Notwendigkeit verfahren wir mit anderen Partitionen genauso und können dann mit geänderten Werten die Partitionen wieder erstellen.

4.7.2.3 GPT schreiben Entspricht die Partitionierung der Festplatte unseren Vorstellungen prüfen wir noch einmal mit dem Befehl **`Verify`** ob alles in Ordnung ist. Werden keine Fehler angezeigt, wählen wir **`Write`** und

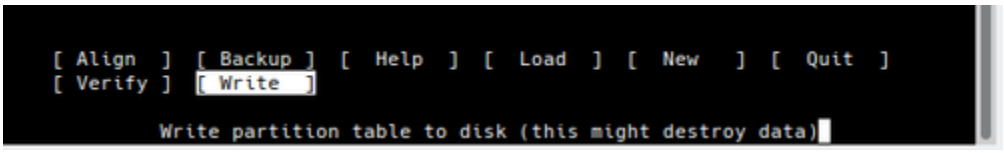


Abbildung 33: Write

dürfen die Sicherheitsabfrage mit **`yes`** beantworten.

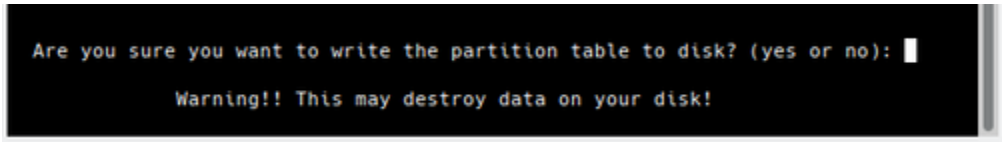


Abbildung 34: Sicherheitsabfrage

Der Warnhinweis sollte ernst genommen werden, denn nach Betätigung der **`Enter`**-Taste verschwinden alle Daten, die zuvor auf der Festplatte waren im Nirwana.

Da cgdisk nur Partitionen, aber keine Dateisysteme erstellt, muss jede der neuen Partitionen formatiert werden. Mit **`Quit`** wird cgdisk beendet.

4.7.3 Formatieren der Partitionen

Wir bleiben im Root-Terminal und lassen uns die Pfade mit den Nummern für jede Partition anzeigen:

```
fdisk -l | grep /dev/sdb
```

Der Befehl generiert die folgende Ausgabe:

```
Disk /dev/sdb: 149,5 GiB, 160041885696 bytes, 312581808 ↙
   sectors
/dev/sdb1       2048    206847    204800  100M EFI System
/dev/sdb2     206848    208895      2048    1M BIOS boot
/dev/sdb3     208896  52637695  52428800   25G Linux root (↙
   x86-64)
/dev/sdb4   52637696  61026303   8388608    4G Linux swap
/dev/sdb5   61026304 260255743 199229440   95G Linux ↙
   filesystem
/dev/sdb6  260255744 312581808  52326064   25G Linux root (↙
   x86-64)
```

Mit diesen Informationen formatieren wir unsere zuvor erstellten Partitionen.

Bitte unbedingt die Manpages **`man mke2fs`**, **`man mkfs.fat`** und **`man mkswap`** lesen.

Die EFI-Systempartition erhält ein **FAT32** Dateisystem.

```
mkfs.vfat /dev/sdb1
```

Die BIOS_Boot-Partition darf nicht formatiert werden!
Sofern der Bootmanager `GRUB` bei der Installation die *“EFI-System”*- und die *“BIOS_Boot”*-Partition findet, benutzt er sie, gleichgültig, welches Installationsziel wir angegeben haben.

Die Linuxpartitionen `sdb3`, `sdb5` und `sdb6` formatieren wir mit **ext4**.

```
mkfs.ext4 /dev/sdb3
```

Die Swap-Partition wird mit:

```
mkswap /dev/sdb4
```

eingerichtet. Danach machen wir sie mit:

```
swapon /dev/sdb4
```

dem System bekannt und kontrollieren, ob der Swap-Speicher verfügbar ist:

```
swapon -s
Filename      Type    Size  Used  Priority
/dev/sdb4           partition 4194304 0     -2
```

Falls Swap korrekt erkannt wurde:

```
swapoff /dev/sdb4
```

Als nächstes ist es unbedingt notwendig, das System neu zu starten, damit das neue Partitionierungs- und Dateisystemschema vom Kernel eingelesen wird.

4.7.4 Booten mit GPT-UEFI oder GPT-BIOS

Falls ein bootbarer Datenträger mit GPT erstellt werden soll, gibt es zwei Möglichkeiten den Bootsektor eines GPT-Datenträgers zu erstellen.

Diese Möglichkeiten sind:

- Der Computer (das Mainboard) besitzt ein UEFI
- UEFI soll zum Booten des GPT-Datenträgers verwendet werden.

oder

- Der Computer (das Mainboard) hat **kein** UEFI sondern ein BIOS. (Alle Mainboard vor 2009 haben kein UEFI)
- Das BIOS soll zum Booten des GPT-Datenträgers verwendet werden.

4.7.4.1 Booten mit UEFI Wenn UEFI zum Booten verwendet werden soll, muss eine mit FAT formatierte *“EFI System”*-Partition (Typ “EF00”) als erste Partition, und eine unformatierte *“BIOS boot”*-Partition (Typ “EF02”) als zweite erstellt werden. Die erste Partition enthält den/die Bootloader.
Während der Installation von siduction wird jegliche Auswahlmöglichkeit der installgui, wohin der Bootloader installiert werden soll, ignoriert, sofern die vorgenannten Partitionen existieren. Der Bootloader von siduction wird in der *“EFI-System”*-Partition unter `/efi/siduction` gespeichert. Die EFI-Systempartition wird auch als `/boot/efi` eingebunden, solange die Option der Einbindung weiterer Partitionen (“mount other partitions”) gewählt ist. Die Einbindung der *“EFI-System”*-Partition muss im Installer nicht extra angegeben werden.

4.7.4.2 Booten mit BIOS Falls das System kein UEFI besitzt, muss als erste eine **BIOS-Boot**-Partition erstellt werden. Diese ersetzt den Sektor eines MBR-partitionierten Datenträgers, der sich zwischen der Partitionierungstabelle und der ersten Partition befindet, und in diesen wird Grub direkt geschrieben.
Die Partition sollte die Größe von 200MB haben. (Der Grund dieser Größe anstelle der konventionellen 32MB liegt darin, um für den Fall eines Wechsels zu UEFI eine ausreichend große Partition zur Verfügung zu haben.)

4.7.5 Erweiterte Befehle von gdisk

gdisk besitzt erweiterte Optionen und Sicherheitsmechanismen die in cgdisk nicht zur Verfügung stehen.

Falls Probleme entdeckt wurden (z. B. überlappende Partitionen oder nicht entsprechende Haupt- und Sicherungspartitionstabellen), besteht die Möglichkeit, diese mit verschiedenen Optionen im Menü `recovery & transformation` zu beheben. Wir starten gdisk mit

```
gdisk  /dev/sdb
```

An der Eingabeaufforderung **`Command (? for help):`** geben wir den Befehl **`r`** ein, um in das Untermenü von *“recovery & transformation”* zu gelangen und anschließend das **`?`**.

```
recovery/transformation command (? for help): ?
b use backup GPT header (rebuilding main)
```

```
c load backup partition table from disk (rebuilding main)
d use main GPT header (rebuilding backup)
e load main partition table from disk (rebuilding backup)
f load MBR and build fresh GPT from it
g convert GPT into MBR and exit
h make hybrid MBR
i show detailed information on a partition
l load partition data from a backup file
m return to main menu
o print protective MBR data
p print the partition table
q quit without saving changes
t transform BSD disklabel partition
v verify disk
w write table to disk and exit
x extra functionality (experts only)
? print this menu
```

Ein drittes Menü, **`experts`** , erreicht man mit **`x`** entweder vom *“main menu”* oder dem *“recovery & transformation menu”*.

```
recovery/transformation command (? for help): x

Expert command (? for help): ?
a set attributes
c change partition GUID
d display the sector alignment value
e relocate backup data structures to the end of the disk
g change disk GUID
i show detailed information on a partition
l set the sector alignment value
m return to main menu
n create a new protective MBR
o print protective MBR data
p print the partition table
q quit without saving changes
r recovery and transformation options (experts only)
s resize partition table
v verify disk
```

```
w write table to disk and exit
z zap (destroy) GPT data structures and exit
? print this menu
```

Dieses Menü ermöglicht Low-Level-Bearbeitung wie Änderung der Partitions GUID oder der GUIDs des Datenträgers (**c** bzw. **g**). Die Option **z** zerstört augenblicklich die GPT-Datenstrukturen. Dies kann sinnvoll sein, wenn der GPT-Datenträger mit einem anderen Partitionierungsschema verwendet werden soll. Falls diese Strukturen nicht ausgelöscht werden, können einige Partitionierungsprogramme wegen des Vorhandenseins von zwei Partitionierungssystemen Probleme haben.

Trotz alledem: die Optionen der Menüs *“recovery & transformation”* und *“experts”* sollten nur benutzt werden, wenn man sich sehr gut mit GPT auskennt. Als “Nicht-Experte” sollte man diese Menüs nur verwenden, wenn ein Datenträger beschädigt ist. Vor jeder drastischen Aktion sollte die Option **b** im Hauptmenü verwendet werden, um eine Sicherungskopie in einer Datei anzulegen und diese auf einem separaten Datenträger speichern. Dadurch kann die originale Konfiguration wieder hergestellt werden, falls die Aktion nicht nach Wunsch läuft.

Zuletzt bearbeitet: 2021-11-28

4.8 Partitionieren mit fdisk

fdisk und **cfdisk** erstellt MBR-Partitionstabellen auf Basis des BIOS. Im Jahr 2000 begann die Einführung von GPT-Partitionstabellen auf Basis des UEFI.

Der neuere Standard **G**lobally Unique Identifier **P**artition **T**able (GPT), der Teil des UEFI-Standards ist, hat bei aktueller Hardware den MBR ersetzt und erlaubt Platten/Partitionen größer als 2 TByte und eine theoretisch unbegrenzte Anzahl primärer Partitionen. Weitere Informationen dazu gibt es in Wikipedia GUID-Partitionstabelle

Wir empfehlen die Partitionierung mit fdisk und cfdisk ausschließlich für ältere Hardware.
Zum Erstellen von GPT-Partitionstabellen bitte die Handbuchseite Partitionieren mit gdisk zu Rate ziehen.

4.8.1 Benennung von Speichergeräten

Bitte BEACHTEN:
siduction verwendet in der `fstab` UUID für die Benennung von Speichergeräten. Bitte das Kapitel Benennung nach UUID zu Rate ziehen.

Festplatten

Informationen über die Geräte erhält man leicht von einem Informationsfenster (Pop-Up), wenn man mit der Maus auf das Icon eines Geräts auf dem Desktop geht. Dies funktioniert sowohl vom Live-ISO als auch bei einem installierten siduction.

Wir empfehlen die Erstellung einer Tabelle (manuell oder generiert), welche die Details aller Geräte enthält. Dies kann sehr hilfreich sein, falls Probleme auftreten. In einem Terminal werden wir mit **`su`** zu **root** und geben **`fdisk -l`** ein. Bei zwei Festplatten bekommen wir z. B. eine Ausgabe ähnlich der unten gezeigten.

```
user1@pc1:/$ su
Passwort:
root@pc1:/# fdisk -l

Disk /dev/sda: 149,5 GiB, 160041885696 bytes, 312581808 ⬋
   sectors
Disk model: FUJITSU MHY2160B
Units: sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x6513a8ff

Device   Boot   Start       End   Sectors Size Id Type
/dev/sda1        2048  41945087  41943040  20G 83 Linux
/dev/sda2    41945088  83888127  41943040  20G 83 Linux
/dev/sda3    83888128  88291327   4403200 2,1G 82 Linux swap
/dev/sda4    88291328 312581807 224290480 107G  5 Extended
/dev/sda5    88293376 249774079 161480704  77G 83 Linux
/dev/sda6   249776128 281233407  31457280  15G 83 Linux
/dev/sda7   281235456 312581807  31346352  15G 83 Linux


Disk /dev/sdb: 119,25 GiB, 128035676160 bytes, 250069680 ↙
   sectors
Disk model: Samsung SSD 850
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x000403b7

Device   Boot  Start       End   Sectors  Size Id Type
/dev/sdb1       2048  17831935  17829888  8,5G 82 Linux swap
/dev/sdb2   17831936 122687487 104855552   50G 83 Linux
/dev/sdb3  122687488 250068991 127381504 60,8G 83 Linux
```

Mit dem Befehl
`fdisk -l > /home/<MEIN USER NAME>/Dokumente/fdisk-l_Ausgabe` erhalten wir eine Text-Datei mit dem gleichen Inhalt.

Partitionen

Die Partitionen auf einer MBR-Festplatte werden durch eine Zahl zwischen 1 und 15 definiert. Es sind maximal 14 mountbare Partitionen möglich.

Es gibt folgende Partitionstypen:
primäre, erweiterte und logische.

Die logischen Partitionen befinden sich innerhalb der erweiterten Partition. Es sind maximal vier primäre bzw. drei primäre und eine erweiterte Partition anlegbar. Die erweiterte Partition wiederum kann bis zu elf logische Partitionen enthalten.
Primäre oder erweiterte Partitionen erhalten eine Bezeichnung zwischen 1 und 4 (zum Beispiel sda1 bis sda4). Logische Partitionen sind immer gebündelt und Teil einer erweiterten Partition. Mit libata können maximal elf logische Partitionen definiert werden, und ihre Bezeichnungen beginnen mit Nummer 5 und enden höchstens mit Nummer 15.

Beispiele

```
4 Partitionen (alles primäre):

|sda1|sda2|sda3|sda4|


6 Partitionen (3 primäre, 1 erweiterte und 3 logische):

|sda1|sda2|sda3|-
                 |         enthält nur
               |sda4| ->   Verweise auf
                 |         logische Partitionen
                 |
               |sda5|sda6|sda7|
```

"/dev/sda5" kann nur eine logische Partition sein (in diesem Fall die erste logische auf diesem Gerät). Sie befindet sich auf der ersten Festplatte des Computers (abhängig von der BIOS-Konfiguration).

"/dev/sdb3" kann nur eine primäre oder erweiterte Partition sein. Der Buchstabe "b" indiziert, dass diese Partition sich auf einem anderen Gerät befindet als die Partition des ersten Beispiels, welche den Buchstaben "a" enthält.

4.8.2 Cfdisk verwenden

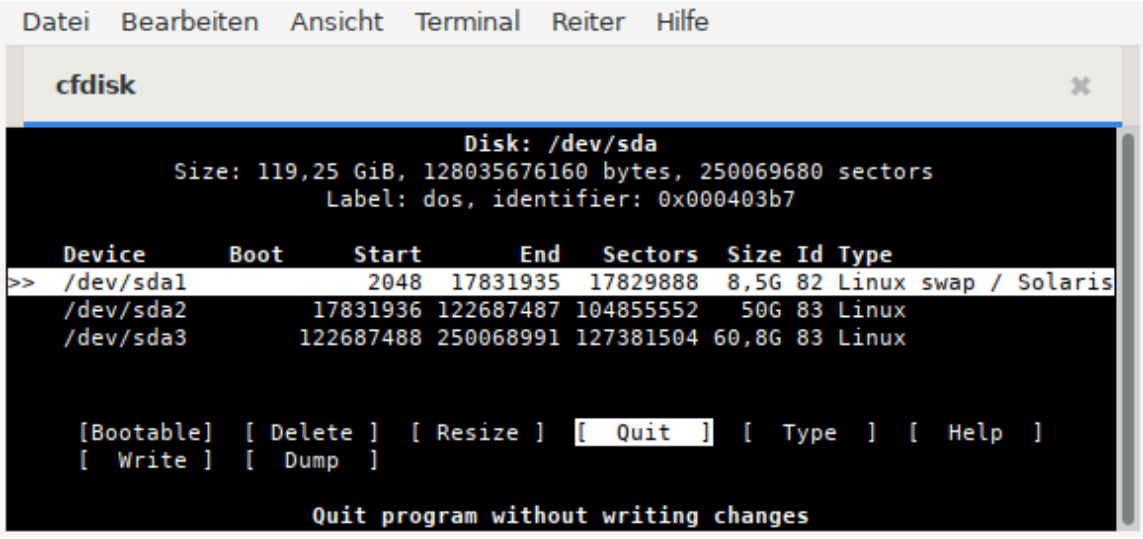
Daten zuvor sichern!
Bei Verwendung jedweder Partitionierungssoftware droht Datenverlust. Daten, die erhalten bleiben sollen immer zuvor auf einem anderen Datenträger sichern.

cfdisk wird in einer Konsole als **root** gestartet (nach **su** ist die Eingabe des root-Passworts gefordert):

```
user1@pc1:/$ su
Passwort:
root@pc1:/#
cfdisk /dev/sda
```

cfdisk nur auf einer Festplatte anwenden, deren sämtliche Partitionen nicht eingehangen sind. Alle Daten gehen mit dem Schreiben der geänderten Partitionstabelle verloren.

4.8.2.1 Die Bedienoberfläche Im ersten Bildschirm zeigt cfdisk die aktuelle Partitionstabelle mit den Namen und einigen Informationen zu jeder Partition. Am unteren Ende des Fensters befinden sich einige Befehlsschalter. Um zwischen den Partitionen zu wechseln, benutzt man die Pfeiltasten **auf** und **ab**, um Befehle auszuwählen, die Pfeiltasten **rechts** und **links**. Mit der **Enter** Taste wird der Befehl ausgeführt.



Wir haben auf der Beispielfestplatte drei Partitionen.

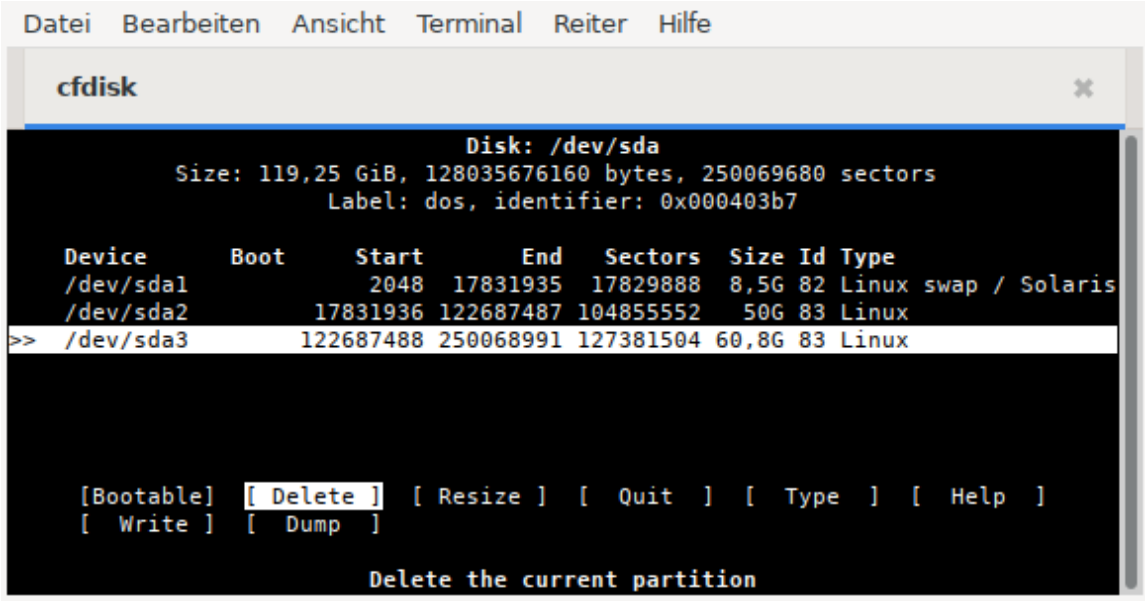
Device	Part. Größe	Part. Typ	Mountpoint
/dev/sda1	8,5G	82 Swap	-
/dev/sda2	50,0G	83 Linux	/
/dev/sda3	60,8G	83 Linux	/Daten

Aus der Daten-Partition möchten wir die Verzeichnisse `/Bilder` und `/Musik` in eigene Partitionen auslagern und dafür mehr Platz schaffen. Gleichzeitig sollen diese auch für ein auf einer weiteren Festplatte residierendes Windows zugänglich sein. Die Root-Partition ist mit 50 GB überdimensioniert und wird verkleinert.

4.8.2.2 Löschen einer Partition

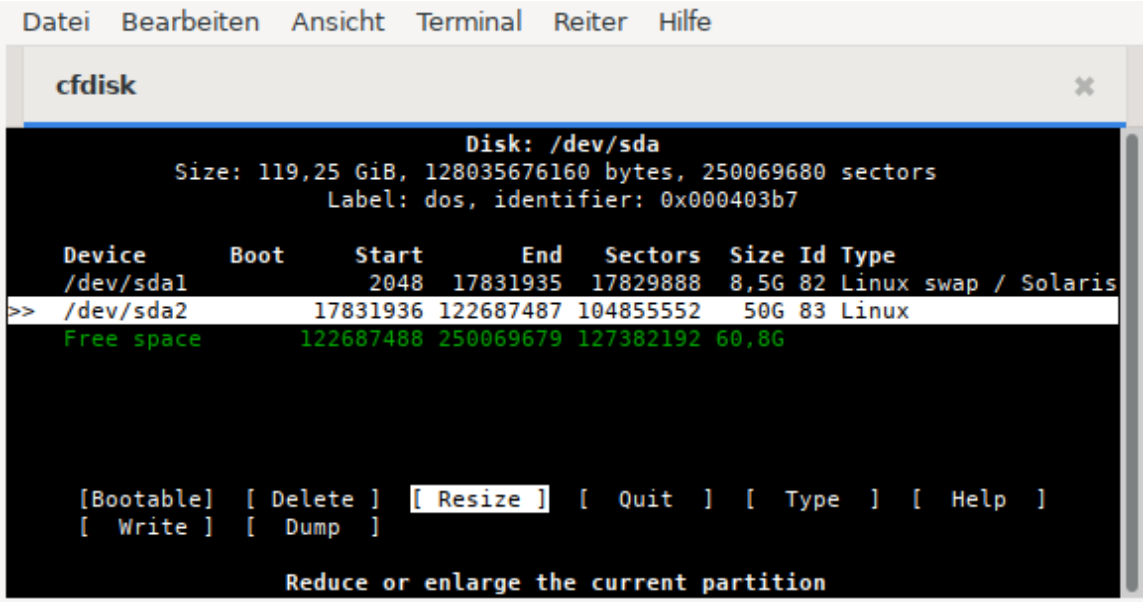
Um Platz zu schaffen, löschen wir die Daten-Partition und verkleinern anschließend die Root-Partition.

Um die Partition `/dev/sda3` zu löschen, wird sie mit den auf-ab-Tasten markiert und der Befehl **`Delete`** mit den Pfeiltasten links-rechts gewählt und durch **`Enter`** bestätigt.

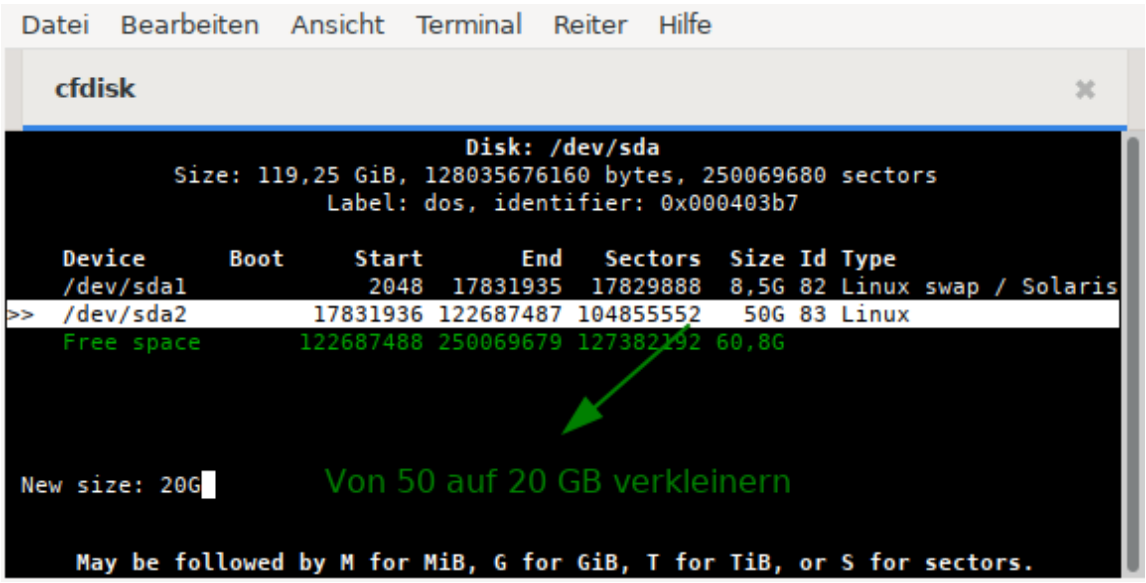


4.8.2.3 Größe einer Partition ändern

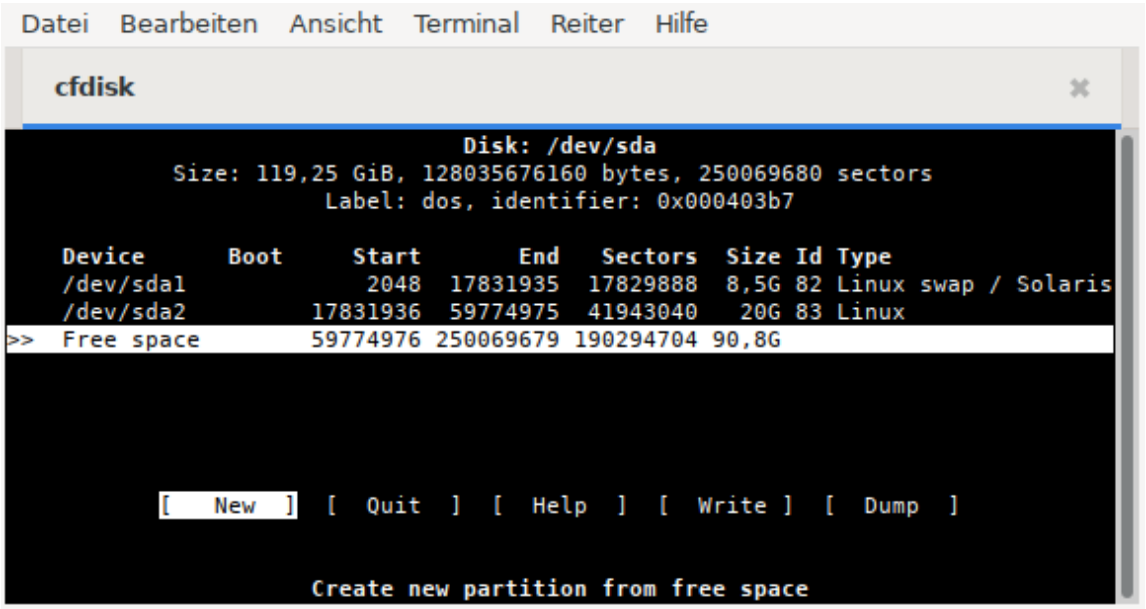
Die Partition `/dev/sda2` wird markiert und der Befehl **`Resize`** ausgewählt und bestätigt.



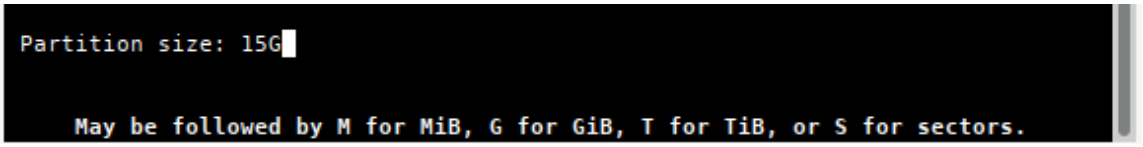
Anschließend erfolgt die Eingabe der neuen Größe von `20G`



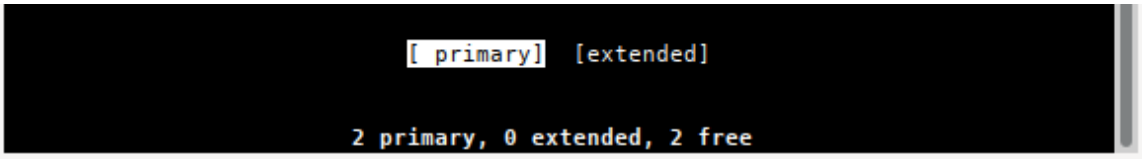
4.8.2.4 Erstellen einer neuen Partition Der nun freie Platz der Festplatte wird markiert. Die Befehlsauswahl springt automatisch auf **`New`**, die zu bestätigen ist.



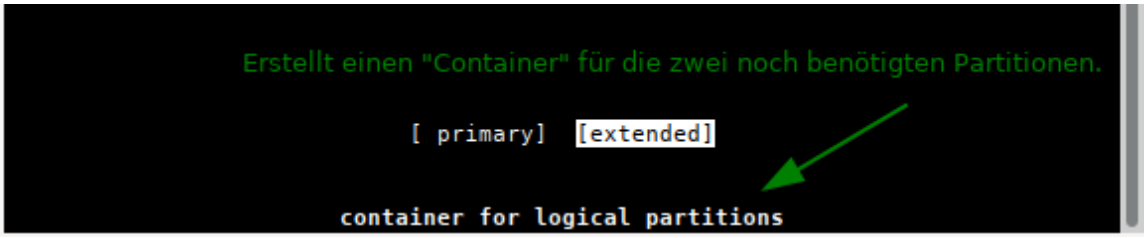
Anschließend ist die neue Größe von `15G` für die Daten-Partition einzugeben.



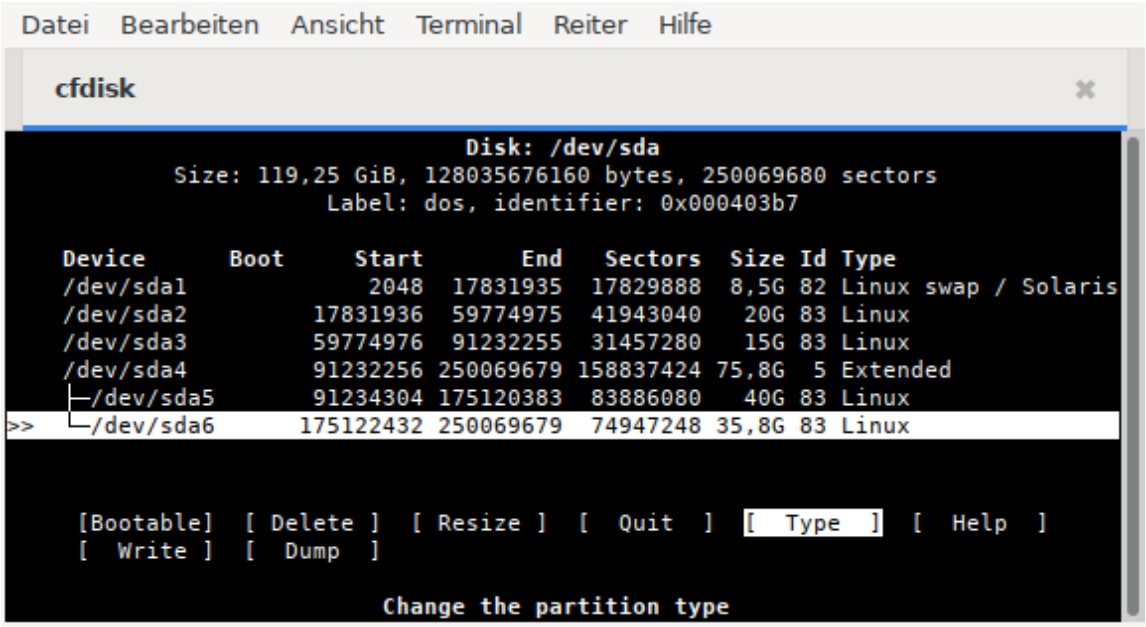
Jetzt muss zwischen einer *“primären”* oder einer *“erweiterten”* (extended) Partition entschieden werden. Wir entscheiden uns für eine primäre Partition.



Danach wird wieder der freie Plattenplatz markiert, bestätigt und die voreingestellte gesamte Größe ebenso bestätigt. In der folgenden Auswahl ist `extended` zu wählen. Dies erstellt die Erweiterte Partition (hier ‘Container’ genannt) in der die zwei zusätzlichen Partitionen anzulegen sind.

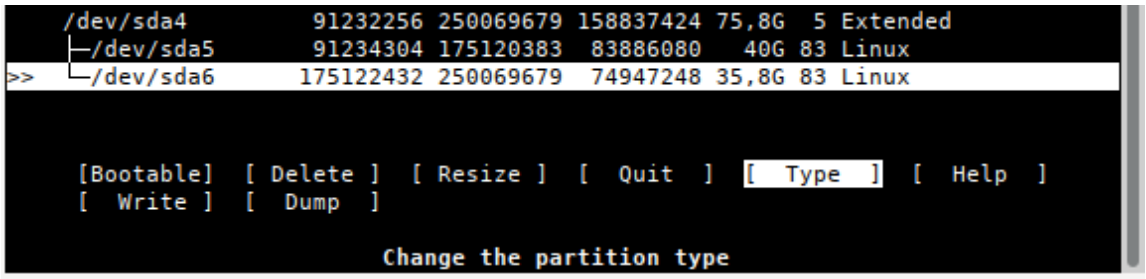


Zum Schluss sind die Partitionen für Musik und Bilder entsprechend dem oben gezeigten Vorgehen in der gewünschten Größe anzulegen. Da nur noch logische Partitionen möglich sind, entfällt die Auswahl zwischen primärer und erweiterter Partition.

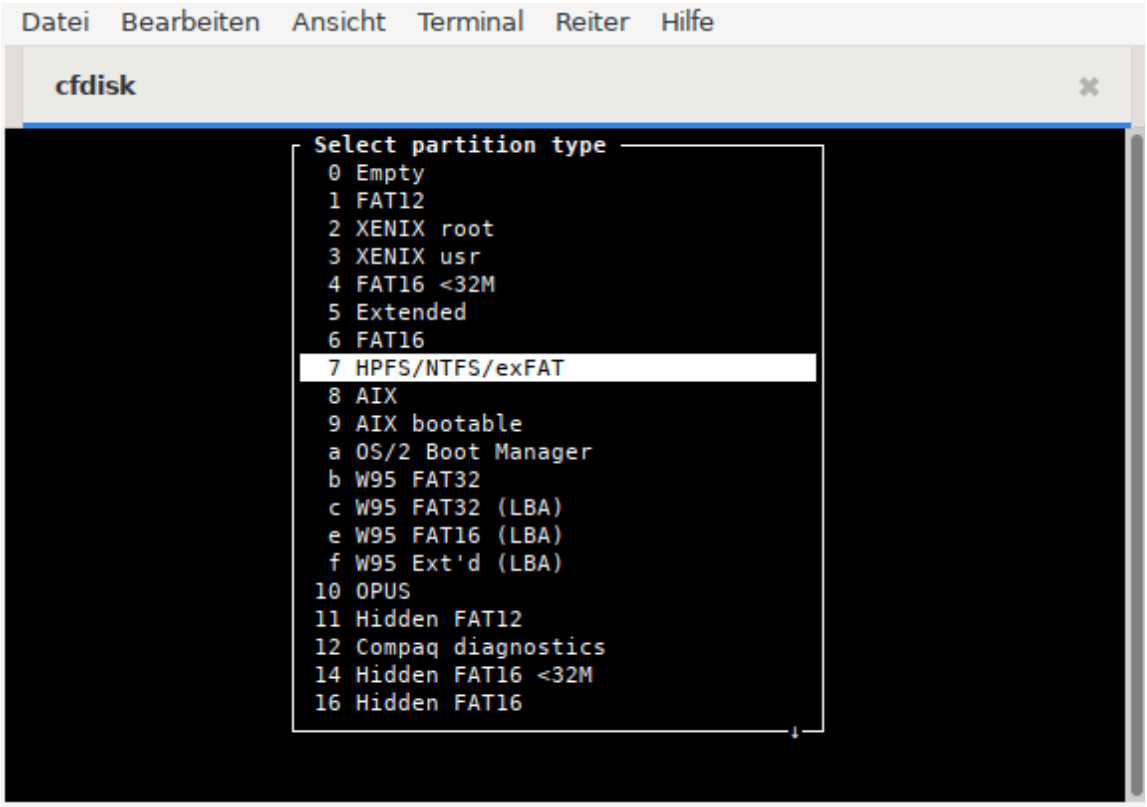


So sieht das Ergebnis aus.

4.8.2.5 Partitionstyp Um den Typ einer Partition zu ändern, markiert man die gewünschte Partition und wählt den Befehl **Type** aus.

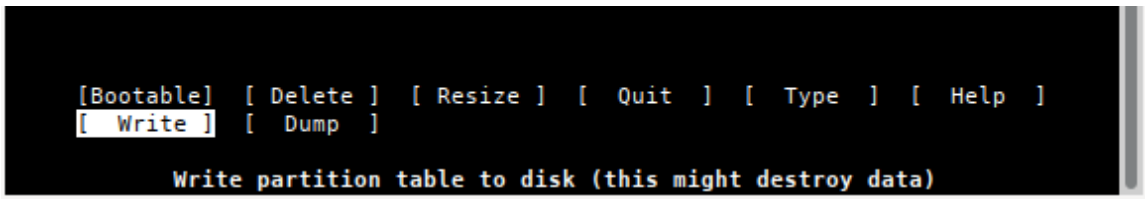


Es erscheint eine Auswahlliste in der mit den Pfeiltasten auf und ab der Partitionstyp gewählt wird. In unserem Beispiel wählen wir für die Partitionen `/dev/sda5` und `/dev/sda6` den Typ *"7 HPFS/NTFS/exFAT"* aus. So kann das oben erwähnte Windows auf die Partition zugreifen.



4.8.2.6 Eine Partition bootfähig machen Für Linux besteht kein Grund, eine Partition bootfähig zu machen, aber einige andere Betriebssysteme brauchen das. Dabei wird die entsprechende Partition markiert und der Befehl **`Bootable`** gewählt (Anmerkung: Bei Installation auf eine externe Festplatte muss eine Partition bootfähig gemacht werden).

4.8.2.7 Partitionstabelle schreiben Wenn alles fertig partitioniert ist, kann das Resultat mit dem Befehl **`Write`** gesichert werden. Die Partitionstabelle wird jetzt auf die Platte geschrieben.



Da damit alle Daten auf der entsprechenden Festplatte/Partition gelöscht werden , sollte man sich seiner Sache wirklich sicher sein, bevor man **`yes`** eintippt und noch einmal mit **`Enter`** bestätigt.

4.8.2.8 Cfdisk beenden Mit dem Befehl **`Quit`** verlassen wir das Programm. Nach Beendigung von cfdisk und vor der Installation sollte man auf jeden Fall rebooten, um die Partitionstabelle neu einzulesen.

4.8.3 Formatieren von Partitionen

Es gibt für Linux verschiedene Filesysteme, die man benutzen kann. Da wären **Ext2**, **Ext4**, **ReiserFs** und für erfahrenere Anwender **XFS**, **JFS** und **ZFS**.
Ext2 kann von Interesse sein, wenn man von Windows aus zugreifen möchte, da es Windows-Treiber für dieses Dateisystem gibt. Ext2-Dateisystem für MS Windows (Treiber und englischsprachige Doku).

Für normalen Gebrauch empfehlen wir das Dateisystem **ext4**. Es ist das Standard-Dateisystem von siduction.

Nach Beendigung von cfdisk wird die Root-Konsole weiter verwendet. Eine Formatierung erfordert Root-Rechte.
Der Befehl lautet **`mkfs.ext4 /dev/sdaX`**. Für “X” trägt man die Nummer der ausgewählten Partition ein.

```
mkfs.ext4 /dev/sda2
mke2fs 1.45.6 (20-Mar-2020)
/dev/sdb2 contains a ext4 file system
  last mounted on Tue May 26 14:26:34 2020
Proceed anyway? (y,N)
```

Die Abfrage wird mit **`y`** beantwortet, wenn man darin sicher ist, dass die richtige Partition formatiert werden soll. Bitte mehrfach überprüfen!

Nach Abschluss der Formatierung muss die Meldung erfolgen, dass **ext4** erfolgreich geschrieben wurde. Ist das nicht der Fall, ist bei der Partitionierung etwas schiefgelaufen oder *“sdaX”* ist keine Linux-Partition. Wir überprüfen mit:

```
fdisk -l /dev/sda
```

Wenn etwas falsch ist, muss gegebenenfalls noch einmal partitioniert werden.

War die Formatierung erfolgreich, so wird dieser Ablauf für die anderen Partitionen wiederholt, wobei der Befehl entsprechend des Partitions-Typ und des gewünsch-

ten Dateisystem anzupassen ist. (z. B.: `mkfs.ext2` oder `mkfs.vfat` oder `mkfs.ntfs` usw.) Bitte die Manpage **`man mkfs`** lesen.

Zuletzt wird die Swap-Partition formatiert, in diesem Fall **sda1**:

```
mkswap /dev/sda1
```

Im Anschluss wird die Swap-Partition aktiviert:

```
swapon /dev/sda1
```

Danach kann in der Konsole überprüft werden, ob die Swap-Partition erkannt wird:

```
swapon -s
```

Bei eingebundener Swap-Partition sollte die Ausgabe auf den vorherigen Befehl etwa so aussehen:

```
Filename        Type        Size      Used   Priority
/dev/sda1       partition   8914940   0      -2
```

Wird die Swap-Partition korrekt erkannt, starten wir den Computer neu.

Jetzt kann die Installation beginnen.

Zuletzt bearbeitet: 2021-03-08

4.9 LVM-Partitionierung - Logical Volume Manager

Es folgt nun eine Basiseinführung. Es liegt am geschätzten Leser, sich tiefer in die Materie einzuarbeiten. Weitere Informationsquellen finden sich am Ende dieses Textes - die Liste erhebt keinen Anspruch auf Vollständigkeit.

Das Arbeiten mit *Logical Volumes* ist viel einfacher als die meisten User glauben. Die beste Eigenschaft von LVM ist, dass Änderungen wirksam werden ohne dafür das System neu starten zu müssen. *Logical Volumes* können mehrere Festplatten umspannen und sind skalierbar. Dies unterscheidet sie von anderen Methoden der Festplattenpartitionierung.

Mit drei Grundbegriffen sollte man vertraut sein:

- **Physisches Volumen (Physical Volume):** Diese sind die physischen, real vorhandenen, Festplatten oder Partitionen wie zum Beispiel `/dev/sda` oder `/dev/sdb1` und werden zum Einbinden/Aushängen verwendet. Mit LVM können mehrere physische Volumen in Volumengruppen zusammengefasst werden.
- **Volumengruppe (Volume Group):** Eine Volumengruppe besteht aus *Physischen Volumen* und ist der Speicherort von *Logischen Volumen*. Eine Volumengruppe kann als “virtuelles Laufwerk” gesehen werden, das aus *Physischen Volumen* zusammengesetzt ist. Zum Verständnis einige Beispiele:
  - Mehrere Speichergeräte (z. B. Festplatten, SSDs, M2-Disks, externe USB-Festplatten usw.) können zu einer Volumengruppe (einem virtuellen Laufwerk) zusammengefasst werden.
  - Mehrere Partitionen eines Speichergerätes können zu einer Volumengruppe (einem virtuellen Laufwerk) zusammengefasst werden.
  - Eine Kombination aus den beiden vorgenannten Möglichkeiten. Z. B. drei SSDs, wovon von der ersten nur zwei Partitionen und die beiden anderen vollständig in der Volumengruppe zusammengefasst werden.
- **Logisches Volumen (Logical Volume):** Logische Volumen werden inerhalb einer *Volumengruppe* erstellt und in das System eingebunden. Man kann sie auch als “virtuelle” Partitionen verstehen. Sie sind dynamisch veränderbar, können in der Größe verändert, neu erstellt, entfernt und verwendet werden.

Ein logisches Volumen kann sich innerhalb der Volumengruppe über mehrere physische Volumen erstrecken.

4.9.1 Sechs Schritte zu Logical Volumes

Achtung
Wir gehen in unserem Beispiel von nicht partitionierten Festplatten aus. Zu beachten ist: Falls alte Partitionen gelöscht werden, gehen alle Daten unwiederbringlich verloren.

Als Partitionierungsprogramm werden cfdisk oder gdisk benötigt, da zur Zeit GParted bzw. der KDE-Partitionsmanager (partitionmanager) das Anlegen von *Logical Volumes* nicht unterstützen. Siehe auch die Handbuchseiten:
Partitionieren mit cfdisk (msdos-MBR)
Partitionieren mit gdisk (GPT-UEFI)

Alle folgenden Befehle und Aktionen erfordern root-Rechte.

1. Erstellung einer Partitionstabelle

```
cfdisk /dev/sda
n    -> erstellt eine neue Partition auf dem Laufwerk
p    -> diese Partition wird eine primäre Partition
1    -> die Partition erhält die Nummer 1 als ↙
   Identifikation
### size allocation  ### setzt den ersten und letzten ↙
   Zylinder auf Default-Werte. Drücke ENTER, um das ↙
   gesamte Laufwerk zu umspannen
t    -> wählt den zu erstellenden Partitionstyp
8e   -> der Hex-Code für eine Linux-LVM
W    -> schreibt Veränderungen auf das Laufwerk.
```

Der Befehl `W` schreibt die Partitionierungstabelle. Falls bis zu diesem Punkt ein Fehler gemacht wurde, kann das vorhandene Partitionierungs-Layout wieder hergestellt werden. Zu diesen Zweck gibt man den Befehl `q` ein, cfdisk beendet sich ohne Schreibvorgang, und alles bleibt wie es zuvor war.

Falls die Volumengruppe mehr als ein Physische Volumen (Laufwerk) umspannen soll, muss obiger Vorgang auf jedem physischen Volumen durchgeführt werden.

2. Erstellen eines physischen Volumens

```
pvcreate /dev/sda1
```

Der Befehl erstellt auf der ersten Partition der ersten Festplatte das physische Volumen.
Dieser Vorgang wird nach Bedarf auf jeder Partition wiederholt.

3. Erstellen einer Volumengruppe

Nun fügen wir die physischen Volumen einer Volumengruppe mit dem Namen **vulcan** hinzu (in unserem Beispiel drei Laufwerke):

```
vgcreate vulcan /dev/sda1 /dev/sdb1 /dev/sdc1
```

Falls dieser Schritt korrekt durchgeführt wurde, kann das Ergebnis in der Ausgabe folgenden Befehls gesehen werden:

```
vgscan
```

vgdisplay zeigt die Größe mit:

```
vgdisplay vulcan
```

4. Erstellung eines logischen Volumens

An dieser Stelle muss entschieden werden, wie groß das logische Volumen zu Beginn sein soll. Ein Vorteil von LVM ist die Möglichkeit, die Größe ohne Reboot anpassen zu können.

In unserem Beispiel wünschen wir uns ein 300GB großes Volumen mit dem Namen **spock** innerhalb der Volumengruppe Namens **vulcan**:

```
lvcreate -n spock --size 300g vulcan
```

5. Formatieren des logischen Volumens

Bitte habe etwas Geduld, dieser Vorgang kann längere Zeit in Anspruch nehmen.

```
mkfs.ext4 /dev/vulcan/spock
```

6. Einbindung des logischen Volumens

   Erstellen des Mountpoints mit

   ```
   mkdir /media/spock/
   ```

   Um das Volumen während des Bootvorgangs einzubinden, muss die Datei `/etc/fstab` mit einem Texteditor angepasst werden.
   Die Verwendung von *"/dev/vulcan/spock"* ist bei einem LVM der Verwendung von UUID-Nummern vorzuziehen, da es damit einfacher ist das Dateisystem zu klonen (keine UUID-Kollisionen). Besonders mit einem LVM können Dateisysteme mit gleicher UUID-Nummer erstellt werden (Musterbeispiel: Snapshots).

   ```
   mcedit /etc/fstab
   ```

   und dann die folgende Zeile entsprechend unseres Beispiels einfügen.

   ```
   /dev/vulcan/spock /media/spock/ ext4 auto,users,rw,exec,↙
      dev,relatime 0 2
   ```

   Optional:
   Der Besitzer des Volumens kann geändert werden, sodass andere Nutzer Lese- bzw. Schreibzugang zum Logical Volumen haben:

   ```
   chown root:users /media/spock
   chmod 775 /media/spock
   ```

Die Schritte 4 bis 6 können wir nun für das neu zu erstellende logische Volumen **kirk** wiederholen.

Ein einfacher LVM sollte nun erstellt sein.

4.9.2 Größenänderung eines Volumens

Wir empfehlen die Verwendung einer Live-ISO, um Partitionsgrößen zu ändern. Obwohl die Vergrößerung einer Partition des laufenden Systems ohne Fehler

durchgeführt werden kann, ist dies bei der Verkleinerung einer Partition nicht der Fall. Anomalien können zu einem Datenverlust führen, vor allem wenn die Verzeichnisse **/** (root) oder **/home** betroffen sind.

Beispiel einer Vergrößerung

Eine Partition soll von 300GB auf 500GB vergrößert werden:

```
umount /media/spock/
```

Erweitern des logischen Volumens:

```
lvextend -L+200g /dev/vulcan/spock
```

Dem Befehl *"lvextend"* ist als Option der Wert für die Größen**änderung** anzugeben und nicht die gewünscht Gesamtgröße.

Anschließend die Größe des Dateisystems ändern:
Der erste Befehl führt zwangsweise eine Check durch, auch wenn das Dateisystem sauber zu sein scheint,
der letzte Befehl hängt das logische Volumen wieder ein.

```
e2fsck -f /dev/vulcan/spock
resize2fs /dev/vulcan/spock
mount /media/spock
```

Beispiel einer Verkleinerung

Eine Partition wird von 500GB auf 280GB verkleinert:

```
umount /media/spock/
```

Die Größe des Dateisystems verringern:

```
e2fsck -f /dev/vulcan/spock
resize2fs /dev/vulcan/spock 280g
```

Danach wird das logische Volumen geändert.

```
lvreduce -L-220g /dev/vulcan/spock
resize2fs /dev/vulcan/spock
mount /media/spock
```

Auch hier ist dem Befehl *“lvreduce”* als Option der Wert für die Größen**änderung** anzugeben.
Der erneute *“resize2sf”*-Befehl passt das Dateisystem exakt an die Größe des logischen Volumens an.

4.9.3 LVM mit einem GUI-Programm verwalten

Gparted bietet die Möglichkeit zur Verwaltung von bereits angelegten Logical Volumes. Das Programm wird als root ausgeführt.

4.9.4 Weitere Infos

- Logical Volume Manager - Wikipedia (Deutsch)
- Working with logical volumes #1 (Englisch)
- Working with logical volumes #2 (Englisch)
- Working with logical volumes #3 (Englisch)
- Größenänderung von Linuxpartitionen - Teil 2 (IBM) (Englisch)

Zuletzt bearbeitet: 2021-05-04

4.10 Das Verzeichnis home verschieben

> Wichtige Information
> Ein existierendes **/home** soll nicht mit einer anderen Distribution verwendet oder geteilt werden, da es bei den Konfigurationsdateien zu Konflikten kommen kann/wird.

Deshalb raten wir generell davon ab eine /home-Partition anzulegen.
Das Verzeichnis `/home` sollte der Ort sein, an dem die individuellen Konfigurationen abgelegt werden, und nur diese. Für alle weiteren privaten Daten sollte eine eigene Datenpartition angelegt, und diese z. B. unter `/Daten` eingehängt werden. Die Vorteile für die Datenstabilität, Datensicherung und auch im Falle einer Datenrettung sind nahezu unermesslich.
Sofern Daten gemeinsam für parallele Installationen bereit stehen sollen, ist diese Vorgehensweise besonders ratsam.

Vorbereitungen

An Hand eines realistischen Beispiels zeigen wir die notwendigen Schritte auf.
Die Ausgangslage:

- Die alte, mittlerweile zu kleine, Festplatte hat die drei Partitionen `/boot/efi`, `/`, `swap`.
- Es existiert bisher noch keine separate Daten-Partition.
- Eine zusätzliche eingebaute Festplatte hat vier Partitionen mit ext4-Dateisystem.
  Davon benutzen wir die Partitionen `sdb4` für die neue Daten-Partition, die wir unter `/Daten` einhängen.

Unsere bisherige `/etc/fstab` hat den Inhalt:

```
$ cat /etc/fstab
...
# <file system>                          <mount point>  <type>
     <options>    <dump><pass>
UUID=B248-1CCA                             /boot/efi   vfat
      umask=0077 0 2
UUID=1c257cff-1c96-4c4f-811f-46a87bcf6abb  /           ext4
      defaults,noatime 0 1
```

```
UUID=2e3a21ef-b98b-4d53-af62-cbf9666c1256  swap        swap ↙
      defaults,noatime 0 2
tmpfs                                      /tmp        tmpfs ↙
     defaults,noatime,mode=1777 0 0
```

Von der zusätzlichen Festplatte benötigen wir die UUID-Informationen. Siehe auch die Handbuchseite Anpassung der fstab.
Der Befehl `blkid` gibt uns Auskunft.

```
$ /sbin/blkid
...
/dev/sdb4: UUID="e2164479-3f71-4216-a4d4-af3321750322" ↙
   BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="000403b7-04"
```

Sicherung des alten /home

Bevor irgendeine Änderung am bestehenden Dateisystem vorgenommen wird, sichern wir als **root** alles unterhalb von `/home` in einem tar-Archiv.

```
# cd /home
# tar cvzpf somewhere/home.tar.gz ./
```

Mountpoint der Daten-Partition

Wir erstellen das Verzeichnis `/Daten` unterhalb `/` und binden die Partition `sdb4` ↙ dort ein. Als Eigentümer und Gruppe legen wir die eigenen Namen fest. Etwas später kopieren wir die privaten Daten, nicht aber die Konfigurationen, aus dem bestehenden `/home` dort hinein.

Mountpoint erstellen und Partition einhängen (als root):

```
# mkdir /Daten
# chown <user>:<group> /Daten
# mount -t ext4 /dev/sdb4 /Daten
```

4.10.1 Private Daten verschieben

Analyse von /home

Wir schauen uns erst einmal unser Home-Verzeichnis genau an.
(Die Ausgabe wurde zur besseren Übersicht sortiert.)

```
~$ ls -la
insgesamt 169
drwxr-xr-x 19 <user> <group> 4096  4. Okt 2020  .
drwxr-xr-x 62 <user> <group> 4096  4. Okt 22:17 ..
-rw-------  1 <user> <group>  330 15. Okt 2020  .bash_history
-rw-r--r--  1 <user> <group>  220  4. Okt 2020  .bash_logout
-rw-r--r--  1 <user> <group> 3528  4. Okt 2020  .bashrc
drwx------ 19 <user> <group> 4096 15. Okt 2020  .cache
drwxr-xr-x 22 <user> <group> 4096 15. Okt 2020  .config
-rw-r--r--  1 <user> <group>   24  4. Okt 2020  .dmrc
drwx------  3 <user> <group> 4096 15. Okt 2020  .gconf
-rw-r--r--  1 <user> <group>  152  4. Okt 2020  .gitignore
drwx------  3 <user> <group> 4096 15. Okt 2020  .gnupg
-rw-------  1 <user> <group> 3112 15. Okt 2020  .ICEauthority
-rw-r--r--  1 <user> <group>  140  4. Okt 2020  .inputrc
drwx------  3 <user> <group> 4096  4. Okt 2020  .local
drwx------  5 <user> <group> 4096 15. Okt 2020  .mozilla
-rw-r--r--  1 <user> <group>  807  4. Okt 2020  .profile
drwx------  2 <user> <group> 4096  4. Okt 2020  .ssh
drwx------  5 <user> <group> 4096 15. Okt 2020  .thunderbird
-rw-------  1 <user> <group>   48 15. Okt 2020  .Xauthority
-rw-------  1 <user> <group> 1084 15. Okt 2020  .xsession-↙
   errors
drwxr-xr-x  2 <user> <group> 4096  4. Okt 2020  Bilder
drwxr-xr-x  2 <user> <group> 4096  4. Okt 2020  Desktop
drwxr-xr-x  2 <user> <group> 4096  4. Okt 2020  Dokumente
drwxr-xr-x  2 <user> <group> 4096  4. Okt 2020  Downloads
drwxr-xr-x  2 <user> <group> 4096  4. Okt 2020  Musik
drwxr-xr-x  2 <user> <group> 4096  4. Okt 2020  Öffentlich
drwxr-xr-x  2 <user> <group> 4096  4. Okt 2020  Videos
drwxr-xr-x  2 <user> <group> 4096  4. Okt 2020  Vorlagen
```

Die Ausgabe zeigt das Home-Verzeichnis kurz nach der Installation mit nur geringfügigen Änderungen.
In den, per default erstellten, Verzeichnissen *“Bilder”* bis *“Vorlagen”* am Ende der Liste, legen wir unsere privaten Dokumente ab. Diese und eventuell zusätzliche, selbst erstellte Verzeichnisse mit privaten Daten, verschieben wir später in die

neue Daten-Partition.
Mit einem Punkt (.) beginnende, “versteckte” Dateien und Verzeichnisse enthalten die Konfiguration und programmspezifische Daten, die wir, von drei Ausnahmen abgesehen, nicht verschieben. Die Ausnahmen sind:
Der Zwischenspeicher *“.cache”*,
der Internetbrowser *“.mozilla”* und
das Mailprogramm *“.thunderbird”*.
Alle drei erreichen mit der Zeit ein erhebliches Volumen und sie enthalten auch viele private Daten. Deshalb wandern sie zusätzlich auf die neue Daten-Partition.

Kopieren der privaten Daten

Zum Kopieren benutzen wir den Befehl `cp` mit der Archiv-Option `-a`, so bleiben die Rechte, Eigentümer und der Zeitstempel erhalten und es wird rekursiv kopiert.

```
~$ cp -a * /Daten/
~$ cp -a .cache /Daten/
~$ cp -a .mozilla /Daten/
~$ cp -a .thunderbird /Daten/
```

Der erste Befehl kopiert alle Dateien und Verzeichnisse, außer die versteckten. Die folgende Ausgabe zeigt das Ergebnis.

```
~$ ls -la /Daten/
insgesamt 45
drwxr-xr-x 13 <user> <group> 4096  4. Mai 2020  .
drwxr-xr-x 20 root     root  4096  4. Okt 2020  ..
drwxr-xr-x  2 <user> <group> 4096  4. Okt 2020  Bilder
drwx------ 19 <user> <group> 4096 15. Okt 2020  .cache
drwxr-xr-x  2 <user> <group> 4096  4. Okt 2020  Desktop
drwxr-xr-x  2 <user> <group> 4096  4. Okt 2020  Dokumente
drwxr-xr-x  2 <user> <group> 4096  4. Okt 2020  Downloads
drwx------  5 <user> <group> 4096 15. Okt 2020  .mozilla
drwxr-xr-x  2 <user> <group> 4096  4. Okt 2020  Musik
drwxr-xr-x  2 <user> <group> 4096  4. Okt 2020  Öffentlich
drwx------  5 <user> <group> 4096 15. Okt 2020  .thunderbird
drwxr-xr-x  2 <user> <group> 4096  4. Okt 2020  Videos
drwxr-xr-x  2 <user> <group> 4096  4. Okt 2020  Vorlagen
```

Die Prüfung der Kopieraktion auf Fehler erfolgt mit dem Befehl **`dirdiff /home/<user>/ /Daten/`**. Es dürfen nur die Dateien und Verzeichnisse gelistet sein, die wir nicht kopiert haben.

Nun befinden sich alle privaten Daten aus dem alten *"/home"* zusätzlich auf der neuen Partition.

Löschen in /home

Für diese Aktion sollten alle Programmfenster, mit Ausnahme des von uns benutzten Terminals, geschlossen werden.
Je nach Desktopumgebung benutzen diverse Anwendungen die per default bei der Installation angelegten Verzeichnisse (z. B. *"Downloads"* oder "*Musik*"). Um den Zugriff dieser Anwendungen auf die Verzeichnisse in der neuen *"/daten"* Partition zu ermöglichen, müssen wir unter `/home/<user>/` die zu löschenden Verzeichnisse durch Link ersetzen.

> Die Befehle vor dem Ausführen bitte genau prüfen, damit nicht aus Versehen etwas falsches gelöscht wird.

```
~$ rm -r Bilder/ && ln -s /Daten/Bilder/ ./Bilder
~$ rm -r Desktop/ && ln -s /Daten/Desktop/ ./Desktop
~$ rm -r Dokumente/ && ln -s /Daten/Dokumente/ ./Dokumente
~$ rm -r Downloads/ && ln -s /Daten/Downloads/ ./Downloads
~$ rm -r Musik/ && ln -s /Daten/Musik/ ./Musik
~$ rm -r Öffentlich/ && ln -s /Daten/Öffentlich/ ./Öffentlich
~$ rm -r Videos/ && ln -s /Daten/Videos/ ./Videos
~$ rm -r Vorlagen/ && ln -s /Daten/Vorlagen/ ./Vorlagen
~$ rm -r .cache/ && ln -s /Daten/.cache/ ./.cache
~$ rm -r .mozilla/ && ln -s /Daten/.mozilla/ ./.mozilla
~$ rm -r .thunderbird/ && ln -s /Daten/.thunderbird/ ./.
   thunderbird
```

Die im /home-Verzeichnis verbliebenen Daten belegen nur noch einen Speicherplatz von weniger als 10 MB.

4.10.2 fstab anpassen

Damit beim Systemstart die neue Daten-Partition eingehangen wird und dem User zur Verfügung steht, muss die Datei `/etc/fstab` geändert werden. Zusätzliche Informationen zur fstab bietet unser Handbuch Anpassung der fstab.
Wir benötigen die oben bereits ausgelesene UUID-Information der Daten-Partition. Zuvor erstellen wir eine Sicherungskopie der fstab mit Datumsanhang:

```
# cp /etc/fstab /etc/fstab_$(date +%F)
# mcedit /etc/fstab
```

Entsprechend unseres Beispiels fügen wir die folgende Zeile in die fstab ein.

```
UUID=e2164479-3f71-4216-a4d4-af3321750322 /Daten ext4 defaults,↙
noatime 0 2
```

Die fstab sollte nun so aussehen:

```
# <file system>                          <mount point>  <type>↙
     <options>    <dump><pass>
UUID=B248-1CCA                              /boot/efi   vfat ↙
      umask=0077 0 2
UUID=1c257cff-1c96-4c4f-811f-46a87bcf6abb  /           ext4 ↙
      defaults,noatime 0 1
UUID=e2164479-3f71-4216-a4d4-af3321750322  /Daten      ext4 ↙
      defaults,noatime 0 2
UUID=2e3a21ef-b98b-4d53-af62-cbf9666c1256  swap        swap ↙
      defaults,noatime 0 2
tmpfs                                       /tmp        tmpfs ↙
     defaults,noatime,mode=1777 0 0
```

Man speichert die Datei mit F2 und beendet den Editor mit F10.

Sollte dennoch irgend etwas schief gehen, so haben wir unsere Daten immer noch im gesicherten tar-Archiv.

Zuletzt bearbeitet: 2021-11-28

5 Netzwerk

Dieser Abschnitt beinhaltet Informationen und Hinweise zur/zum

- Network Manager Kommandline Tool, seiner Bedienung, welche Geräte vorhanden und konfiguriert sind, wie Verbindungen hergestellt und getrennt, und wie von WLAN zu LAN und zurück gewechselt werden kann.
- IWD, Intels wireless daemon, als Ersatz für den WPA-Supplicant; seine Installation, Bedienung und die Zusammenarbeit mit dem Network Manager.
- SAMB Client einrichten um auf Windows-Freigaben zugreifen zu können.
- SSH einrichten und absichern
- LAMP-Testserver für Entwickler (lokal), seinen Komponenten und deren Installation, wo die Logdateien zu finden sind und wie man eventuell auftretende Fehler und korrigiert.
- LAMP - Apache, seine Verzeichnisse im Dateisystem, die Konfiguration, das Benutzer- und Rechte-Management, den sicheren Betrieb als lokalen Server und die Verwendung von HTTPS.
- LAMP - MariaDB, ihre Verzeichnisse im Dateisystem, die Erstkonfiguration, das Kommandline Interface “mariadb”, phpMyAdmin und die Integration in systemd.
- LAMP - PHP, die Verzeichnisse im Dateisystem, die PHP-Unterstützung für Apache2, die Konfiguration, die Installation und das Handling von Modulen und wo die Logdateien zu finden sind.

Zuletzt bearbeitet: 2021-11-29

5.1 Network Manager Kommandline Tool

Allgemeine Hinweise

Der Networkmanager ist mittlerweile in allen graphischen Oberflächen von siduction integriert und größtenteils selbsterklärend. Er ersetzt die im Terminal benutzten Netzwerkkommandos `ifup`, `ifdown` bzw. `ifconfig`. Das Vorurteil das sich der Networkmanager nicht für die Kommandozeile eignet oder gar instabil läuft gehört ins Reich der Märchenwelt. Steht keine graphische Oberfläche zur Verfügung, oder wird die Komandozeile bevorzugt, existiert mit `nmcli` ein leistungsfähiger Kommandozeilenclient für den täglichen Gebrauch des Networkmanagers.

In den nachfolgenden Beispielen gehen wir von zwei konfigurierten Verbindungen aus. Eine WLAN-Verbindung (Name: Einhorn_2, Interface wtx7ckd90b81bbd, (früher; wlan)) und einer kabelgebundenen Verbindung (Name: Kabelgebundene Verbindung 1, Interface evp0s3f76 (früher: eth0)). Bitte die Verbindungsnamen an eure Gegebenheiten anpassen.

Installation des Network Managers

Falls der Networkmanager auf dem System nicht installiert ist, kann man dies nachholen. Im nachfolgenden Kommando sind alle Pakete die man braucht um alle möglichen Verbindungsarten zu konfigurieren (mobiles Breitband, WLAN und LAN Verbindungen), sowie das grafische KDE-Plasma-Widget für den NM. Bitte alles in eine Zeile eingeben.

```
apt update
apt install network-manager modemmanager mobile-broadband-↙
   provider-info network-manager-pptp
 plasma-nm network-manager-vpnc network-manager-openvpn
```

5.1.1 Network Manager verwenden

Die Eingaben können sowohl in einem virtuellen Terminal (Tastenkombination **`Str`**+**`Umschalt`**+**`F2`**) als auch in der graphischen Oberfläche in einer Konsole getätigt werden. In den abgebildeten Beispielen wurden die Angaben aus Datenschutzgründen abgeändert.

Konfigurierte Verbindungen anzeigen

Mit dem Kommando **`nmcli c`** können die konfigurierten Verbindungen, die man am System angelegt hat, angezeigt werden.

```
nmcli c
NAME                        UUID                                  TYPE      DEVICE
WirelessAdapter_2           4c247331-05bd-4ae6-812b-6c70b35dc348  wifi      wtx7ckd90b81bbd
Kabelgebundene Verbindung 1 847d4195-3355-33bc-bea8-7a016ab86824  ethernet  evp0s3f76
Kabelgebundene Verbindung 2 efc70b04-01f1-31fc-b948-5fd9ceca651d  ethernet  --
MobilesNetzUMTS             fe0933bc-f5fa-4b94-8622-d03c4195721e  gsm       xyz72905dg34
```

Abbildung 35: nmcli c

Im obigen Beispiel sind vier Verbindungen vorhanden WLAN, 2x LAN und eine Mobile Breitbandverbindung.

Informationen zu WIFI Netzen anzeigen

Welche WLAN-Netze sind überhaupt am Standort verfügbar, das kann man sich in kompakter Form mit **`nmcli dev wifi list`** anzeigen lassen.

```
nmcli dev wifi list
IN-USE  BSSID              SSID          MODE   CHAN  RATE        SIGNAL  BARS  SECURITY
*       14:CF:20:C6:1A:8F  WLAN-01       Infra  6     270 Mbit/s  92      ▂▄▆█  WPA2
        54:67:64:3D:02:30  WLAN-02       Infra  1     405 Mbit/s  85      ▂▄▆█  WPA2
        D0:AA:2A:17:EE:9B  WLAN-03       Infra  11    270 Mbit/s  52      ▂▄__  WPA2
```

Abbildung 36: nmcli dev wifi list

Konfigurierte Geräte anzeigen

Will man wissen welche Geräte (Interfaces) überhaupt dem Networkmanager bekannt sind ist **`nmcli d`** hilfreich.

```
nmcli d
DEVICE           TYPE      STATE            CONNECTION
evp0s3f76        ethernet  verbunden        Kabelgebundene Verbindung 1
wtx7ckd90b81bbd  wifi      verbunden        Einhorn_2
evp3u3           ethernet  nicht verfügbar  --
ttyACM0          gsm       nicht verbunden  --
```

Abbildung 37: nmcli d

Sehr detaillierte Informationen (Eigenschaften) gibt es mit **`nmcli dev show`** zu den eigenen verfügbaren Verbindungen. Hier nur der Auszug für das WLAN.

Die Zugangsdaten zum WLAN kann man sich mit **`nmcli dev wifi show`** anzeigen lassen.

```
nmcli dev show
[...]
GENERAL.DEVICE:                         wtx7ckd90b81bbd
GENERAL.TYPE:                           wifi
GENERAL.HWADDR:                         7C:FA:83:C2:6B:BD
GENERAL.MTU:                            1500
GENERAL.STATE:                          100 (verbunden)
GENERAL.CONNECTION:                     Einhorn_2
GENERAL.CON-PATH:                       /org/freedesktop/NetworkManager/ActiveConnection/2
IP4.ADDRESS[1]:                         192.168.0.6/24
IP4.GATEWAY:                            192.168.0.1
IP4.ROUTE[1]:                           dst = 0.0.0.0/0, nh = 192.168.0.1, mt = 600
IP4.ROUTE[2]:                           dst = 192.168.0.0/24, nh = 0.0.0.0, mt = 600
IP4.DNS[1]:                             192.168.0.1
IP4.DOMAIN[1]:                          home
IP6.ADDRESS[1]:                         2a02:810d:cc0:c4c:7edd:90ff:feb2:1bbd/64
IP6.ADDRESS[2]:                         fe80::7edd:90ff:feb2:1bbd/64
IP6.GATEWAY:                            fe80::362c:c4ff:fe17:1bf1
IP6.ROUTE[1]:                           dst = 2a02:810d:cc0:c4c::/64, nh = ::, mt = 256
IP6.ROUTE[2]:                           dst = fe80::/64, nh = ::, mt = 256
IP6.ROUTE[3]:                           dst = ::/0, nh = fe80::dc53:e2ff:fe81:6d46, mt = 1024
IP6.ROUTE[4]:                           dst = ::/0, nh = fe80::362c:c4ff:fe17:1bf1, mt = 1024
IP6.ROUTE[5]:                           dst = ff00::/8, nh = ::, mt = 256, table=255
[...]
```

Abbildung 38: nmcli dev show

```
nmcli dev wifi show
SSID: Einhorn_2
Sicherheit: WPA
Passwort: das steht jetzt nicht hier
```

Abbildung 39: nmcli dev wifi show

Der zusätzlich generierte QR-Code vereinfacht den Login für Smartphone und Tablet.

Verbindungen wechseln

Um eine Verbindungsart zu wechseln, z.B. von LAN auf eine WLAN Verbindung, muss man die bestehende aktive Verbindung abbauen und die neue aktivieren. Hier muss man definitiv das Interface angeben, da ein `nmcli con down id <Name>` zwar funktioniert, die Verbindung, wenn es eine Systemverbindung ist, aber sofort wieder aufgebaut wird.

Um die automatische Verbindung zu verhindern hilft der Befehl **`nmcli dev disconnect <Schnittstellenname>`**.
Zuerst beenden wir die LAN-Verbindung und fragen danach den Status ab.

```
# nmcli dev disconnect evp0s3f76
Gerät »evp0s3f76« wurde erfolgreich getrennt.
# nmcli dev status
DEVICE           TYPE      STATE            CONNECTION
evp0s3f76        ethernet  nicht verbunden  --
wtx7ckd90b81bbd  wifi      nicht verbunden  --
evp3u3           ethernet  nicht verfügbar  --
ttyACM0          gsm       nicht verbunden  --
```

Jetzt die WLAN Verbindung aktivieren mit **`nmcli con up id <Verbindungsname>`**:

```
# nmcli con up id Einhorn_2
Verbindung wurde erfolgreich aktiviert
# nmcli dev status
DEVICE           TYPE      STATE            CONNECTION
wtx7ckd90b81bbd  wifi      verbunden        Einhorn_2
evp0s3f76        ethernet  nicht verbunden  --
evp3u3           ethernet  nicht verfügbar  --
ttyACM0          gsm       nicht verbunden  --
```

Man kann das Ganze noch in eine Kommandozeile packen, dann wird der Wechsel sofort durchgeführt.

Von LAN zu WLAN:

```
nmcli dev disconnect evp0s3f76 && sleep 2 && nmcli con up id ↙
   Einhorn_2
```

Umgekehrt von WLAN zu LAN:

```
nmcli dev disconnect wtx7ckd90b81bbd && sleep 2 && nmcli con ↙
   up id 'Kabelgebundene Verbindung 1'
```

5.1.2 Weiterführende Informationen

- Die Manpage **`man nmcli`**
- Ubuntuusers Wiki

Zuletzt bearbeitet: 2021-05-22

5.2 IWD

Intels iNet wireless daemon (iwd) schickt den wpa-supplicant in den wohlverdienten Ruhestand. Nur ein Zehntel so groß und viel schneller, ist iwd der Nachfolger. iwd funktioniert alleine oder zusammen mit NetworkManager, systemd-networkd und Connman.
Zwei Dinge, die iwd noch nicht kann, sind die Verbindung per WPA/WPA2 Enterprise und der korrekte Umgang mit versteckten Netzwerken. Bei solchen Arbeitsumgebungen sollte man bei wpa-supplicant bleiben oder ab siduction 2021.3.0 wieder zum wpa-supplicant wechseln.

Weiterführende Informationen bietet das Arch Linux wiki bzw. das debian wiki.

Seit siduction 2021.3.0 kommt iwd als Standard für den Verbindungsaufbau im WLAN zum Einsatz. Unsere Implementation läuft mit NetworkManager.

Seit siduction 2021.1.0 wurde iwd bereits in den Flavours Xorg und noX ausgeliefert. Wer möchte, kann iwd in den anderen Flavours installieren. Siehe weiter unten: IWD statt wpa_supplicant.

Vor siduction 2021.1.0: Auch bei einem etwas älteren Snapshot kann iwd installiert werden (getestet mit siduction 2018.3.0 und linux-image-5.15.12-1-siduction-amd64). Bitte ebenfalls der Anleitung unter IWD statt wpa_supplicant folgen.

5.2.1 Grafische Konfigurationsprogramme

- **NetworkManager** Für den NetworkManager gibt es verschiedene grafische Oberflächen z.B. für den plasma-desktop/kde `plasma-nm` oder für gnome `network-manager-gnome` und andere. Ihr Benutzung sollte selbsterklärend sein!
- **conman** ist ein von Intel entwickelter Netzwerkmanager, klein und Ressourcen schonend ist, mehr dazu im Arch-Wiki
- **iwgtk** ist nicht in debian-Quellen, es muss aus dem Sourcecode gebaut werden und ist auf github zu finden.

5.2.2 Konfiguration im Terminal

iwd und NetworkManager

1. Der schnellste und einfachste Weg iwd mit dem NetworkManager zu nutzen ist eine Konsole zu öffnen und diesen Befehl einzugeben:

   ```
   ~$ nmtui
   ```

   Es startet im Terminal eine textbasierte, graphische Oberfläche des NetworkManagers. Die Bedienung sollte selbsterklärend sein!

2. Das Kommandline-tool `nmcli` des NetworkManagers benutzen. Ausführliche Informationen hierzu finden sie auf unserer Handbuchseite Network Manager im Terminal

   Ich beschreibe hier nur kurz den schnellsten Weg ein Netzwerk mit Hilfe des NetworkManagers in der Kommandozeile einzurichten. Vorausgesetzt man hat alle Informationen, reicht jener Einzeiler:

   ```
   ~$ nmcli dev wifi con "<ssid>" password password name "<name>"
   ```

   "ssid" bezeichnet den Namen des Netzwerkes

   Zum Beispiel:

   ```
   nmcli dev wifi con "HomeOffice" password W1rkl1chS3hrG3h31m name "HomeOffice"
   ```

iwd standalone (ohne NetworkManager)

Intels iwd bringt ein eigenes Kommandline-tool Namens `iwctl` mit. Bitte iwctl nur benutzen, wenn der NetworkManager und wpa_supplicant deinstalliert oder beide im systemd maskiert wurden.

Als erstes sollte die Hilfe zu iwctl aufgerufen werden, um zu sehen was alles möglich ist. Dafür geben wir im Terminal den Befehl `iwctl` ein, dann am Eingabe-Prompt `help`.

```
[iwd]# help

                            iwctl version 1.12
--------------------------------------------------------------------------------
  Usage
--------------------------------------------------------------------------------
  iwctl [--options] [commands]

                            Available options
--------------------------------------------------------------------------------
  Options                                               Description
--------------------------------------------------------------------------------
  --username                                            Provide username
  --password                                            Provide password
  --passphrase                                          Provide passphrase
  --dont-ask                                            Don't ask for missing
                                                        credentials
  --help                                                Display help

                            Available commands
--------------------------------------------------------------------------------
  Commands                                              Description
--------------------------------------------------------------------------------

Adapters:
  adapter list                                          List adapters
  adapter <phy> show                                    Show adapter info
  adapter <phy> set-property <name> <value>             Set property

Ad-Hoc:
  ad-hoc list                                           List devices in Ad-hoc mode
  ad-hoc <wlan> start <"network name"> <passphrase>     Start or join an existing
                                                        Ad-Hoc network called
                                                        "network name" with a
                                                        passphrase
```

Um heraus zu finden welche WiFi Schnittstelle wir nutzen geben wir folgenden Befehl ein.

```
[iwd]# device list
                      Devices                                  *
--------------------------------------------------------------------
  Name    Address             Powered   Adapter   Mode
--------------------------------------------------------------------
  wlan0   00:01:02:03:04:05   on        phy0      station
```

In diesem Falle ist es *“wlan0”* und es läuft (*“Powered on”*) im *“station”* mode.

Nun scannen wir nach einem aktiven Netzwerk

```
[iwd]# station wlan0 scan
[iwd]# station wlan0 get-networks
```

Jetzt können wir uns zu unserem Netzwerk verbinden.

```
[iwd]# station wlan0 connect SSID
```

“SSID” bezeichnet den Namen des Netzwerkes.

Es wird noch das Passwort abgefragt und wir sollten mit unserem Netzwerk verbunden sein, dies können wir mit *“station list”* oder *“station wlan0 get-networks”* Nachprüfen.

```
[iwd]# station list
            Devices in Station Mode
--------------------------------------------
  Name         State          Scanning
--------------------------------------------
  wlan0        connected
```

Das ganze kann mit folgendem Befehl abgekürzt werden, so man alle nötigen Informationen hat!

```
iwctl --passphrase <passphrase> station device connect SSID
```

Zum Beispiel:

```
~$ iwctl --passphrase W1rkl1chS3hrG3h31m station wlan0 ↙
   connect HomeOffice
```

5.3 IWD statt wpa_supplicant

Allen, die iwd als Ersatz für wpa_supplicant mit einem etwas älteren Snapshot als siduction 2021.3.0 nutzen möchten, richten sich bitte nach den folgenden Anweisungen.

5.3.1 IWD installieren

> Anmerkung:
> Es ist möglich, dass nicht freie Firmware von einem USB-Stick oder via LAN installiert werden muss!
> Debian installiert bei Verwendung von apt den NetworkManager immer zusammen mit wpa_supplicant.

Um den wpa_supplicant los zu werden, gibt es zwei Möglichkeiten, wobei die zweite Möglichkeit die sinnvollere und einfachere ist.

1. NetworkManager ohne apt aus den Sourcen installieren.
2. Den wpa_supplicant.service im systemd nicht starten bzw. maskieren.
   Da siduction systemd nutzt, werden wir nicht darauf eingehen, wie iwd ohne systemd konfiguriert wird!

Möchte man iwd ohne NetworkManager nutzen, so muss man sich darüber keine Gedanken machen, sondern den NetworkManager und wpa_supplicant mit

```
~# apt purge network-manager wpasupplicant
```

mitsamt seiner Konfiguration von der Platte putzen.

Vorgehensweise bei installiertem NetworkManager und iwd < 1.21-2

1. *“iwd”* installieren.
2. *“NetworkManager.service”* stoppen.
3. *“wpa_supplicant.service”* stoppen und maskieren.
4. Die Datei `/etc/NetworkManager/conf.d/nm.conf` angelegen und iwd dort eingetragen.
5. Die Datei `/etc/iwd/main.conf` anlegen und mit entsprechendem Inhalt befüllen.
6. *“iwd.service”* aktivieren und starten.
7. *“NetworkManager.service”* wieder starten.

Jetzt einfach die folgenden Befehle als root im Terminal ausführen, um iwd zu nutzen:

```
~# apt update
```

```
~# apt install iwd
~# systemctl stop NetworkManager.service
~# systemctl disable --now wpa_supplicant.service
~# echo -e '[device]\nwifi.backend=iwd' > /etc/NetworkManager↙
   /conf.d/nm.conf
~# touch /etc/iwd/main.conf
~# echo -e '[General]\nEnableNetworkConfiguration=true\n\n[↙
   Network]\nNameResolvingService=systemd' > /etc/iwd/main.↙
   conf
~# systemctl enable --now iwd.service
~# systemctl start NetworkManager.service
```

Schauen ob es geklappt hat

Wir lassen uns die beiden Konfigurationsdateien anzeigen.

- /etc/NetworkManager/conf.d/nm.conf

```
~$ cat /etc/NetworkManager/conf.d/nm.conf
[device]
wiFi.backend=iwd
```

- /etc/iwd/main.conf

```
~$ cat /etc/iwd/main.conf
[General]
EnableNetworkConfiguration=true

[Network]
NameResolvingService=systemd
```

Vorgehensweise bei installiertem NetworkManager und iwd >= 1.21-2

Ab Version 1.21-2 bringt iwd eine eigene Konfigurationsdatei `/etc/iwd/main↙.conf` mit. Die Vorgehensweise gleicht der eben genannten mit der Ausnahme, dass wir die Konfigurationsdatei nicht mehr anlegen, sondern in ihr das Kommentarzeichen vor “EnableNetworkConfiguration=true” entfernen.

Bitte die folgenden Befehle als root im Terminal ausführen:

```
~# apt update
~# apt install iwd
~# systemctl stop NetworkManager.service
~# systemctl disable --now wpa_supplicant.service
~# echo -e '[device]\nwifi.backend=iwd' > /etc/NetworkManager/conf.d/nm.conf
~# sed -i -E 's/#(EnableNetworkConfiguration=true)/\1/' /etc/iwd/main.conf
~# systemctl enable --now iwd.service
~# systemctl start NetworkManager.service
```

Schauen ob es geklappt hat

Wir lassen uns die beiden Konfigurationsdateien anzeigen.

- /etc/NetworkManager/conf.d/nm.conf

```
~$ cat /etc/NetworkManager/conf.d/nm.conf
[device]
wiFi.backend=iwd
```

- /etc/iwd/main.conf

```
~$ cat /etc/iwd/main.conf

[...]
[General]
# iwd is capable of performing network configuration on its own, including
# DHCPv4 based address configuration.  By default this behavior is
# disabled, and an external service such as NetworkManager, systemd-network
# or dhcpclient is required.  Uncomment the following line if you want iwd
# to manage network interface configuration.
#
EnableNetworkConfiguration=true
#
[...]
```

Jetzt ist man in der Lage, sich im Terminal mit den oben beschriebenen Befehlen **nmtui**, **nmcli** oder **iwctl** WiFi Hardware anzeigen zu lassen, sie zu konfigurieren und sich mit einem Netzwerk zu verbinden.
Oder man benutzt den NetworkManager in der graphischen Oberfläche. Siehe: Grafische Konfigurationsprogramme

5.3.2 Zurück zum wpa_supplicant

Vorausgesetzt der NetworkManager und wpa_supplicant sind installiert, benötigen wir folgende Arbeitsschritte:

1. *“NetworkManager.service”* stoppen.
2. *“iwd.service”* stoppen und maskieren.
3. Die Datei `/etc/NetworkManger/conf.d/nm.conf` umbenennen.
4. *“wpa_supplicant.service”* demaskieren und starten.
5. *“NetworkManager.service”* wieder starten.

```
~# systemctl stop NetworkManager.service
~# systemctl disable --now iwd.service
~# mv /etc/NetworkManager/conf.d/nm.conf /etc/NetworkManager/↙
   conf.d/nm.conf~
~# systemctl unmask wpa_supplicant.service
~# systemctl enable --now wpa_supplicant.service
~# systemctl start NetworkManager.service
```

Jetzt wird wieder der wpa_supplicant für die Verbindung mit der WiFi-Hardware benutzt.

Zuletzt bearbeitet: 2022-01-13

5.4 SAMBA

5.4.1 Client-Konfiguration

um mit siduction über das Netzwerk auf Windows-Freigaben zugreifen zu können

- Alle Befehle werden in einem Terminal oder einer Konsole als **root** ausgeführt.
- Die URL wird in Dolphin als normaler **user** aufgerufen .
  server = Servername oder IP der Windows-Maschine
  share = Name der Freigabe

  Im KDE-Dateimanager Dolphin wird die URL folgendermaßen eingeben: `smb://server` oder mit dem gesamten Pfad: `smb://server/share`

In einer Konsole können die Freigaben auf einem Server damit gesehen werden:

```
smbclient -L server
```

Um eine Freigabe in einem Verzeichnis sehen zu können (mit Zugriff für ALLE User), muss ein Einhängeort (Mountpoint) existieren. Ist das nicht der Fall, muss ein Verzeichnis als Einhängepunkt erstellt werden (der Name ist beliebig):

```
mkdir -p /mnt/server_share
```

Eine Freigabe wird mit diesem Befehl eingehängt:

```
mount -t cifs -o username=Administrator,uid=$UID,gid=$GID //↙
   server/share /mnt/server_share
```

sollte es hier zu einer Fehlermeldung kommen, dann kann das an der verwendeten SMB Protokoll-Version liegen. In Debian wird SMB 1.0 aus Sicherheitsgründen nicht mehr benutzt. Leider gibt es auch heute noch Systeme, welche SMB nur per Version 1.0 bereit stellen. Um auf solch eine Freigabe zugreifen zu können, wird als Mountoption dann noch `vers=1.0` benötigt. Der komplette Befehl lautet dann

```
mount -t cifs -o username=Administrator,vers=1.0,uid=$UID,gid↙
   =$GID //server/share /mnt/server_share
```

Eine Verbindung wird mit diesem Befehl beendet:

```
umount /mnt/server_share
```

Um einen Samba-Share automatisch einzubinden, kann die Datei `/etc/fstab` nach folgendem Muster ergänzt werden:

```
//server/share /mnt/server_share cifs noauto,x-systemd.
   automount,x-systemd.idle-timeout=300,\
user=username,password=**********,uid=$UID,gid=$GID   0 0
```

Es ist aber nicht empfehlenswert, das Passwort im Klartext in die fstab zu schreiben. Als bessere Variante erzeugt man `~.smbcredentials` mit folgendem Inhalt an:

```
username=<benutzer>
password=<passwort>
```

Der resultierende Eintrag für /etc/fstab ist dann

```
//server/share /mnt/server_share cifs noauto,x-systemd.
   automount,x-systemd.idle-timeout=300,\
credentials=</pfad/zu/.smbcredentials>,uid=$UID,gid=$GID 0 0
```

Die Variablen *"UID"* und *"GID"* entsprechen denen des users, dem das Share gegeben werden soll. Man kann aber auch `uid=<username>` und `gid=<groupname>` schreiben.

5.4.2 siduction als Samba-Server

Natürlich kann man in siduction auch einen SMB-Server stellen. Die Einrichtung als Samba-Server hier im Handbuch zu beschreiben würde den Rahmen allerdings sprengen. Das Internet hält viele HowTo zu diesem Thema bereit.

Unsere Empfehlungen:

Thomas Krenn, Einfache Samba Freigabe unter Debian
Debian, Windows Freigaben mit Samba

Es finden sich noch viele weitere Seiten zu diesem Thema im Netz.

Zuletzt bearbeitet: 2022-04-20

5.5 SSH

Definition von SSH aus Wikipedia :

Secure Shell oder SSH bezeichnet sowohl ein Netzwerkprotokoll als auch entsprechende Programme, mit deren Hilfe man auf eine sichere Art und Weise eine verschlüsselte Netzwerkverbindung mit einem entfernten Gerät herstellen kann. Häufig wird diese Methode verwendet, um sich eine entfernte Kommandozeile quasi auf den lokalen Rechner zu holen, das heißt, auf der lokalen Konsole werden die Ausgaben der entfernten Konsole ausgegeben und die lokalen Tastatureingaben werden an den entfernten Rechner gesendet. Hierdurch wird der Effekt erreicht, als säße man vor der entfernten Konsole, was beispielsweise sehr gut zur Fernwartung eines in einem entfernten Rechenzentrum stehenden Root-Servers genutzt werden kann. Die neuere Protokoll-Version SSH-2 bietet weitere Funktionen wie Datenübertragung per SFTP.

Die IANA hat dem Protokoll den TCP-Port 22 zugeordnet, jedoch lassen sich in den Konfigurationsdateien des Daemons auch beliebige andere Ports auswählen, um z.B. Angriffe zu erschweren, da der SSH-Port dem Angreifer nicht bekannt ist.

5.5.1 SSH absichern

Es ist nicht sicher, Root-Anmeldung via SSH zu erlauben. Es gilt, Anmeldungen als Root nicht zum Standard zu machen, denn Debian sollte sicher sein, nicht unsicher. Ebenso sollen Angreifer nicht die Möglichkeit haben, über zehn Minuten einen wortlistenbasierten Passwort Angriff (brute force attack) auf den SSH-Login durchzuführen. Deshalb ist es sinnvoll, das Zeitfenster der Anmeldung sowie die Anzahl möglicher Versuche einzuschränken.

Um SSH sicherer zu machen, verwendet man einen Texteditor, und bearbeitet die Datei `/etc/ssh/sshd_config`.

Folgende Einstellungen können zur Erhöhung der Sicherheit angepasst werden:

- `Port <gewünschter Port>:`
  Dieser Eintrag muss auf den Port verweisen, der auf dem Router zur Weiterleitung freigeschaltet ist. Wenn nicht bekannt ist, was gemacht werden soll, soll der Einsatz von SSH zur Remote Steuerung noch einmal überdacht werden. Debian setzt den Port 22 als Standard. Es ist jedoch ratsam, einen Port

außerhalb des Standardscanbereichs zu verwenden, deswegen verwenden wir z.B. Port 5874:

```
Port 5874
```

- `ListenAddress <IP des Rechners oder der Netzwerkschnittstelle>:`
  Da der Port vom Router weitergeleitet wird, muss der Rechner eine statische IP-Adresse benutzen, sofern kein lokaler DNS-Server verwendet wird. Aber wenn etwas so Kompliziertes wie SSH unter Benutzung eines lokalen DNS-Servers aufgesetzt werden soll und diese Anweisungen benötigt werden, kann sich leicht ein gravierender Fehler einschleichen. Wir verwenden eine statische IP für das Beispiel:

  ```
  ListenAddress 192.168.2.134
  ```

  Protokoll 2 ist bereits Grundeinstellung bei Debian, aber man sollte sicher sein und daher nochmals überprüfen.

- `LoginGraceTime <Zeitrahmen des Anmeldevorgangs>:`
  Die erlaubte Zeitspanne beträgt als Standard absurde 600 Sekunden. Da man für gewöhnlich keine zehn Minuten benötigt, um Benutzernamen und Passwort einzugeben, stellen wir eine etwas vernünftigere Zeitspanne ein:

  ```
  LoginGraceTime 45
  ```

  Nun hat man 45 Sekunden Zeit zum Anmelden, und Hacker haben keine zehn Minuten bei jedem Versuch, das Passwort zu knacken.

- `PermitRootLogin <yes>:`
  Warum Debian hier Erlaubnis zur Anmeldung als Root erteilt, ist nicht nachvollziehbar. Wir korrigieren zu ‘no’:

  ```
  PermitRootLogin no
  StrictModes yes
  ```

- `MaxAuthTries <Anzahl der erlaubten Anmeldungsversuche>:`
  Mehr als 3 oder 4 Versuche sollten nicht ermöglicht werden:

  ```
  MaxAuthTries 3
  ```

Folgende Einstellungen müssen hinzugefügt werden, so sie nicht vorhanden sind:

- `AllowUsers <xxx>:`
  Benutzernamen, welchen der Zugriff via SSH erlaubt ist, getrennt durch Leerzeichen. Nur eingetragene Benutzer können den Zugang verwenden, und dies nur mit Benutzerrechten. Mit `adduser` sollte man einen User hinzufügen, der speziell zur Nutzung von SSH verwendet wird:

  ```
  AllowUsers werauchimmer1 werauchimmer2
  ```

- `PermitEmptyPasswords <xxx>:`
  dem Benutzer soll ein schönes langes Passwort gegeben werden, das man in einer Million Jahren nicht erraten kann. Dieser Benutzer sollte der einzige mit SSH Zugriff sein. Ist er einmal angemeldet, kann er mit `su` Root werden:

  ```
  PermitEmptyPasswords no
  ```

- `PasswordAuthentication <xxx>:`
  natürlich muss hier 'yes' gesetzt werden. Es sei denn, man verwendet einen KeyLogin.

  ```
  PasswordAuthentication yes [wenn man keine keys verwendet
     ]
  ```

Schlussendlich:

```
systemctl restart ssh
```

Nun hat man eine etwas sichere SSH-Konfiguration. Nicht vollkommen sicher, nur besser, vor allem wenn man einen Benutzer hinzugefügt hat, der speziell zur Verwendung mit SSH dient.

5.5.2 SSH für X-Window Programme

ssh -X ermöglicht die Verbindung zu einem entfernten Computer und die Anzeige von dessen Grafikserver X auf dem eigenen lokalen Computer. Den Befehl gibt man als Benutzer (nicht als Root) ein (und man beachte, dass X ein Großbuchstabe ist):

```
$ ssh -X username@xxx.xxx.xxx.xxx (or IP)
```

Man gibt das Passwort für den Benutzernamen des entfernten Computers ein und startet eine graphische Anwendung in der Shell. Beispiele:

```
$ iceweasel ODER oocalc ODER oowriter ODER kspread
```

Bei sehr langsamen Verbindungen kann es von Vorteil sein, die Komprimierungsoption zu nutzen, um die Übertragungsrate zu erhöhen. Bei schnellen Verbindungen kann es jedoch zum entgegengesetzten Effekt kommen:

```
$ ssh -C -X username@xxx.xxx.xxx.xxx (or IP)
```

Weitere Informationen:

```
$man ssh
```

Anmerkung: Falls ssh eine Verbindung verweigert und man eine Fehlermeldung erhält, sucht man in $HOME nach dem versteckten Verzeichnis `.ssh` , löscht die Datei `known_hosts` und versucht einen neuen Verbindungsaufbau. Dieses Problem tritt hauptsächlich auf, wenn man die IP-Adresse dynamisch zugewiesen hat (DCHP).

5.5.3 Kopieren scp via ssh

scp ist ein Befehlszeilenprogramm (Terminal/CLI), um Dateien zwischen Netzwerkcomputern zu kopieren. Es verwendet ssh zur Authentifizierung und zum sicheren Datentransfer, daher verlangt scp zur Anmeldung ein Passwort bzw. eine Passphrase.

So man ssh-Rechte an einem Netzwerk-PC oder Netzwerk-Server besitzt, ermöglicht scp das Kopieren von Partitionen, Verzeichnissen oder Dateien zu oder von einem Netzwerkcomputer (bzw. zu einem Bereich auf selbigem), für den man Zugangsrechte besitzt. Dies kann zum Beispiel ein PC oder Server im lokalen Netzwerk sein oder aber auch ein Computer in einem fremden Netzwerk oder ein lokales USB-Laufwerk. Der Kopiervorgang kann zwischen entfernten Computern/-Speichergeräten stattfinden.

Es können rekursiv auch ganze Partitionen bzw. Verzeichnisse mit dem Befehl `scp -r` kopiert werden. Zu beachten ist, dass scp -r auch symbolischen Links im Verzeichnisbaum folgt.

Beispiele:

1. Kopieren einer Partition:

```
scp -r <user>@xxx.xxx.x.xxx:/media/disk1part6/ /media/↙
   diskXpartX/
```

2. Kopieren eines Verzeichnisses auf einer Partition, in diesem Fall eines Verzeichnisses mit der Bezeichnung “photos” im $HOME:

```
scp -r <user>@xxx.xxx.x.xxx:~/photos/ /media/diskXpartX/↙
   xx
```

3. Kopieren einer Datei in einem Verzeichnis einer Partition, in diesem Fall eine Datei im $HOME:

```
scp <user>@xxx.xxx.x.xxx:~/filename.txt /media/diskXpartX↙
   /xx
```

4. Kopieren einer Datei auf einer Partition:

```
scp <user>@xxx.xxx.x.xxx:/media/disk1part6/filename.txt /↙
   media/diskXpartX/xx
```

5. Falls man sich im Laufwerk bzw. Verzeichnis befindet, in das ein Verzeichnis bzw. eine Datei kopiert werden soll, verwendet man nur einen **.** (Punkt):

```
scp -r <user>@xxx.xxx.x.xxx:/media/disk1part6/filename.↙
   txt .
```

Weitere Informationen bietet die Manpage:

```
man scp
```

5.5.4 SSH mit Dolphin

Sowohl Dolphin als auch Krusader sind fähig, auf Daten eines entfernten Rechners zuzugreifen, indem sie das *“sftp”* Protokoll benutzen, welches in ssh vorhanden ist.

So wird es gemacht:
1) Man öffnet ein neues Dolphin-Fenster
2) Die Syntax in der Adress-Leiste ist: “sftp://username@ssh-server.com”.

Beispiel 1: ein Dialog-Fenster öffnet sich und fragt nach dem SSH-Passwort. Man gibt das Passwort ein und klickt auf OK:

```
sftp://siduction1@remote_hostname_or_ip
```

Beispiel 2: es wird nicht nach einem Passwort gefragt, man wird direkt verbunden.

```
sftp://username:password@remote_hostname_or_ip
```

Für eine LAN-Umgebung

```
sftp://username@10.x.x.x
oder
sftp://username@198.x.x.x
```

Bitte richtige IP eingeben! Anschließend folgt ein Dialog-Fenster zur Eingabe des ssh-Passworts: Dieses eingeben und auf OK klicken.
Eine SSH-Verbindung im Dolphin ist nun hergestellt. In diesem Dolphin-Fenster kann man mit den Dateien auf dem SSH-Server arbeiten, als wären es lokale Dateien.

ANMERKUNG: wenn ein anderer Port als 22 (Grundeinstellung) benutzt wird, muss dieser bei Verwendung von sftp angegeben werden:

```
sftp://user@ip:port
```

“user@ip:port” - dies ist die Standardsyntax für viele Protokolle/Programme wie sftp und smb.

5.5.5 SSHFS - auf einem entfernten Computer mounten

SSHFS ist eine einfache, schnelle und sichere Methode unter Verwendung von FUSE, um ein entferntes Dateisystem einzubinden. Auf Serverseite benötigt man ausschließlich einen laufenden ssh-daemon.

Auf Seite des Clients muss vermutlich sshfs erst installiert werden:

```
apt update && apt install sshfs
```

fuse3 und *groups* sind bereits auf dem ISO und müssen nicht extra installiert werden.

Das Einbinden eines entfernten Dateisystems ist sehr einfach:

```
sshfs -o idmap=user username@entfernter_hostname:verzeichnis ↙
   lokaler_mountpunkt
```

Wenn kein bestimmtes Verzeichnis angegeben wird, wird das Home-Verzeichnis des entfernten Nutzers eingebunden. Bitte beachten: der Doppelpunkt "`:`" ist unbedingt erforderlich, auch wenn kein Verzeichnis angegeben wird!

Nach erfolgter Einbindung verhält sich das entfernte Verzeichnis wie jedes andere lokale Dateisystem. Man kann wie auf einem lokalen Dateisystem nach Dateien suchen, diese lesen und ändern sowie Skripte ausführen.

Die Einbindung des entfernten Hosts wird mit folgendem Befehl gelöst:

```
fusermount -u lokaler_mountpunkt
```

Bei regelmäßiger Nutzung von sshfs empfiehlt sich ein Eintrag in /etc/fstab:

```
sshfs#remote_hostname://remote_directory /local_mount_point ↙
   fuse -o idmap=user ,allow_other,uid=1000,gid=1000,noauto,↙
   fsname=sshfs#remote_hostname://remote_directory 0 0
```

Als nächstes muss das Kommentarzeichen vor *"user_allow_other"* in der Datei `/etc/fuse.conf` entfernt werden:

```
# Allow non-root users to specify the 'allow_other' or '↙
   allow_root'
# mount options.
```

```
#
user_allow_other
```

Dies ermöglicht jedem Nutzer der Gruppe fuse, das Dateisystem einzubinden bzw. zu lösen:

```
mount /pfad/zum/mount/punkt # Einbindung
umount /pfad/zum/mount/punkt # Lösen
```

Mit diesem Befehl prüft man, ob man Mitglied der Gruppe fuse ist:

```
cat /etc/group | grep fuse
```

Die Antwort sollte in etwa so aussehen:

```
fuse:x:117: <nutzername>
```

Falls der Nutzername (username) nicht gelistet ist, verwendet man als root den Befehl adduser:

```
adduser <nutzername> fuse
```

Zur Beachtung: Der Benutzer wird erst nach einem neuerlichen Einloggen Mitglied der Gruppe “fuse” sein.
Jetzt sollte der gewünschte Nutzername gelistet und folgender Befehl ausführbar sein:

```
mount lokaler_mountpunkt
    und
umount lokaler_mountpunkt
```

Zuletzt bearbeitet: 2021-11-29

5.6 LAMP-Webserver

Ein lokaler Testserver für Entwickler

Das Akronym **LAMP** bezieht sich auf eine Reihe freier Software, die gemeinsam genutzt wird, um dynamische Webseiten zu betreiben:

- **L**inux: Betriebssystem
- **A**pache: Web-Server
- **M**ariaDb: Datenbank-Server (ab Debian 9 ‘Stretch’, zuvor mySQL)
- **P**HP, Perl und/oder Python: Skriptsprachen

Verwendungsmöglichkeiten als Server:

1. **ein lokaler Testserver für Webdesigner ohne Internetverbindung (siehe dieses Kapitel)**
2. ein privater (Daten-)Server mit Internetverbindung
3. ein privater Webserver mit umfassender Internetverbindung
4. ein kommerzieller Webserver

Unser Ziel ist es, einen LAMP-Testserver für Entwickler aufsetzen, der über LAN direkt mit dem Arbeitsplatz-PC verbunden ist. Darüber hinaus soll es aus Gründen der Sicherheit für den Server keine Verbindung zu einem lokalen Netzwerk oder gar zum Internet geben. Einzige Ausnahme: Der Server wird temporär und ausschließlich für System- und Software- Aktualisierungen über eine zweite Netzwerkschnittstelle mit dem Internet verbunden.

Zur Beachtung:
Der Desktop-PC, mit dem täglich gearbeitet wird, soll nicht als Server dienen. Als Server soll ein eigener PC verwendet werden, der ansonsten keine weiteren Aufgaben erfüllt.

Im Server-PC sollte mindestens 500MB RAM Arbeitsspeicher zur Verfügung stehen. Weniger RAM wird Probleme bereiten, da ein Server mit MariaDB/MySQL viel RAM benötigt, um ansprechend zu laufen.

Die zu installierenden Pakete sind:

```
apache2
mariadb-server
mariadb-client
php
php7.4-mysql
phpmyadmin
```

Wie bei siduction üblich, erledigen wir die Installationen im `multi-user.target (init 3)` im Terminal.

Vorbereitungen

Falls der Kommandozeilenbrowser *w3m* noch nicht installiert wurde, holen wir das jetzt nach:

```
# apt update
# apt install w3m
```

Das ermöglicht es uns Apache und PHP sofort im Terminal zu testen und erst nach Abschluss aller notwendigen Installationen wieder in die graphische Oberfläche zurückzukehren.

Nun räumen wir noch apt auf.
Der Befehl `apt autoremove` sollte zu der folgenden Ausgabe führen. Wenn nicht, bestätigen wir das Entfernen nicht mehr benötigter Pakete mit `j`.

```
#apt autremove
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut.
Statusinformationen werden eingelesen.... Fertig
0 aktualisiert, 0 neu installiert, 0 zu entfernen und 0 nicht
    aktualisiert.
```

Diese Maßnahme erleichtert uns im Fall einer fehlerhaften Installation die Reparatur ganz wesentlich.
Siehe unten Troubleshooting

Es ist sinnvoll sich bereits vor der Installation einige Daten zu notieren.

Während der Installation notwendig:

- Ein **Passwort** für den Datenbankbenutzer **root** in phpMyAdmin.

Später, für die Konfiguration notwendig:

- **Apache**
  - Server Name
  - Server Alias
  - IP-Adresse des Servers
  - Name des PC
  - IP-Adresse des PC
- **MariaDB:**
  - Den *Namen* der Datenbank die für das Entwicklungsprojekt verwendet werden soll.
  - Den *Namen* (Login-Name) eines neuen Datendank-Benutzers für das Entwicklungsprojekt.
  - Das *Passwort* für den neuen Datendank-Benutzer.
  - Den *Namen* (Login-Name) eines neuen Datenbank Administrators.
  - Das *Passwort* für den Datenbank Administrator.

5.6.1 Apache installieren

Die Installation des Webservers Apache erfordert nur die beiden folgenden Befehle. Der install-Befehl holt sich noch die zusätzlichen Pakete *apache2-data* und *apache2-utils* herein. Anschließend fragen wir den Status von Apache ab und testen gleich die Start- und Stopanweisungen.

```
# apt update
# apt install apache2
[...]
Die folgenden NEUEN Pakete werden installiert:
  apache2 apache2-data apache2-utils
[...]
```

```
Möchten Sie fortfahren? [J/n] j
[...]

# systemctl status apache2.service●
 apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; ⇙
        enabled; vendor preset: enabled)
     Active: active (running) since Sun 2020-12-06 14:24:44 ⇙
        CET; 4min 8s ago
[...]
```

Wie zu erkennen ist, wurde Apache sofort aktiviert.

```
# systemctl stop apache2.service
# systemctl status apache2.service●
 apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; ⇙
        enabled; vendor preset: enabled)
     Active: inactive (dead) since Sun 2020-12-06 14:30:27 ⇙
        CET; 6s ago
[...]

# systemctl start apache2.service
# systemctl status apache2.service●
 apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; ⇙
        enabled; vendor preset: enabled)
     Active: active (running) since Sun 2020-12-06 14:30:59 ⇙
        CET; 3s ago
[...]
```

Der Apache Webserver ist geladen und lässt sich problemlos handhaben. Jetzt prüfen wir seine Funktion mit:

```
w3m http://localhost/index.html
```

Die Apache-Begrüßungsseite mit *“It works!”* erscheint.
Wir beenden w3m mit `q` und bestätigen mit `y`.

Als **ServerRoot** wird das Verzeichnis `/etc/apache2/` bezeichnet. Es enthält die Konfiguration.
Als **DocumentRoot** wird das Verzeichnis `/var/www/html/` bezeichnet. Es enthält die Dateien der Webseite.

Für weitere Informationen und Hinweise zur Absicherung bitte die Handbuchseite LAMP-Apache lesen.

5.6.2 MariaDB installieren

Die Installation von MariaDB gestaltet sich ähnlich einfach in dem die Metapakete “mariadb-server” und “mariadb-client” angefordert werden.

```
# apt install mariadb-server mariadb-client
[...]
Die folgenden NEUEN Pakete werden installiert:
  galera-4 libcgi-fast-perl libcgi-pm-perl libdbd-mariadb-↙
     perl libfcgi-perl libhtml-template-perl libmariadb3
  mariadb-client mariadb-client-10.5 mariadb-client-core-10.5↙
      mariadb-common mariadb-server mariadb-server-10.5
  mariadb-server-core-10.5 mysql-common socat
[...]
Möchten Sie fortfahren? [J/n] j
```

Weitere Informationen zu MariaDB und der Konfiguration liefert unser Handbuch in LAMP-MariaDB

5.6.3 PHP installieren

Zur Installation der Scriptsprache PHP genügt der Befehl:

```
# apt install php
[...]
Die folgenden NEUEN Pakete werden installiert:
  apache2-bin libapache2-mod-php7.4 libaprutil1-dbd-sqlite3 ↙
     libaprutil1-ldap php
  php-common php7.4 php7.4-cli php7.4-common php7.4-json php7↙
     .4-opcache php7.4-readline
[...]
```

```
Möchten Sie fortfahren? [J/n] j
```

Wie schon zuvor, holt das Metapaket eine ganze Reihe von Abhängigkeiten zusätzlich herein.
Um nach der Installation zu prüfen, ob php korrekt läuft, wird die Datei `info.php` in `/var/www/html` mit der Funktion phpinfo() auf die Art erstellt, wie es hier angegeben ist:

```
mcedit /var/www/html/info.php
```

Den folgenden Text einfügen

```
<?php
phpinfo();
?>
```

mit `F2` speichern, `F10` beendet mcedit.

Danach wird der Terminal-Browser w3m dorthin gelinkt:

```
w3m http://localhost/info.php
oder
w3m http://yourip:80/info.php
```

```
PHP logo

PHP Version 7.4.11

System       Linux <hostname> 5.9.13-towo.1-siduction-amd64
   ...
Build Date   Oct 6 2020 10:34:39
server API   Apache 2.0 Handler
...
```

Erhalten wir eine Ausgabe, die wie oben gezeigt beginnt und alle php-Konfigurationen und Grundeinstellungen enthält, so funktioniert PHP und benutzt als *“server API”* den *“Apache 2.0 Handler”*.

Wir beenden w3m mit `q` und bestätigen mit `y`.

Jetzt fehlt noch die Unterstützung für MariaDB/mysql in PHP. Wir benötigen das PHP-Modul *php7.4-mysql*.

```
# apt install php7.4-mysql
```

Wenn wir jetzt wieder die Seite http://localhost/info.php aufrufen, finden wir im Bereich der Module (sie sind alphabetisch sortiert) die Einträge zu *“mysqli”* und *“mysqlnd”*.

Weitere Informationen zu der Konfiguration von PHP und der Verwaltung ihrer Module enthält die Handbuchseite LAMP-PHP

5.6.4 phpMyAdmin installieren

Um die Datenbank MariaDB zu administrieren benötigen wir *phpmyadmin*:

```
# apt install phpmyadmin
[...]
Die folgenden NEUEN Pakete werden installiert:
  dbconfig-common dbconfig-mysql icc-profiles-free libjs-⇙
     openlayers libjs-sphinxdoc libjs-underscore libonig5 ⇙
     libzip4
  php-bacon-qr-code php-bz2 php-dasprid-enum php-gd php-⇙
     google-recaptcha php-mbstring php-mysql
  php-phpmyadmin-motranslator php-phpmyadmin-shapefile php-⇙
     phpmyadmin-sql-parser php-phpseclib php-psr-cache
  php-psr-container php-psr-log php-symfony-cache php-symfony⇙
     -cache-contracts php-symfony-expression-language
  php-symfony-service-contracts php-symfony-var-exporter php-⇙
     tcpdf php-twig php-twig-extensions php-xml php-zip
  php7.4-bz2 php7.4-gd php7.4-mbstring php7.4-xml php7.4-zip ⇙
     phpmyadmin
0 aktualisiert, 38 neu installiert, 0 zu entfernen und 60 ⇙
   nicht aktualisiert.
Es müssen noch 15,7 MB von 15,8 MB an Archiven ⇙
   heruntergeladen werden.
Nach dieser Operation werden 70,9 MB Plattenplatz zusätzlich ⇙
   benutzt.
Möchten Sie fortfahren? [J/n]  j
```

Während der Installation erscheinen die zwei Dialoge.
Im ersten, zu Beginn, wählen wir *“apache2”* und bestätigen mit `ok`

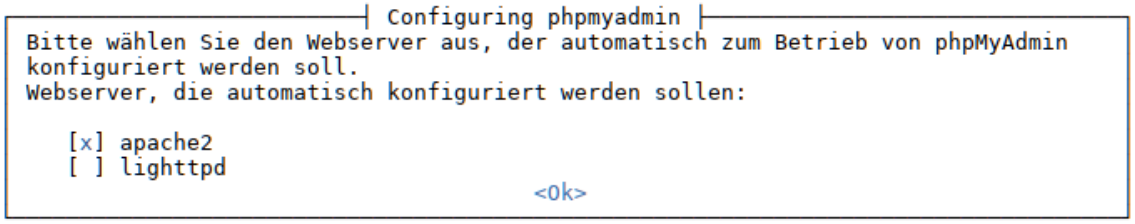


Abbildung 40: PHPMyAdmin Webserverauswahl

im zweiten, am Ende der Installation, wählen wir *“ja”* aus.

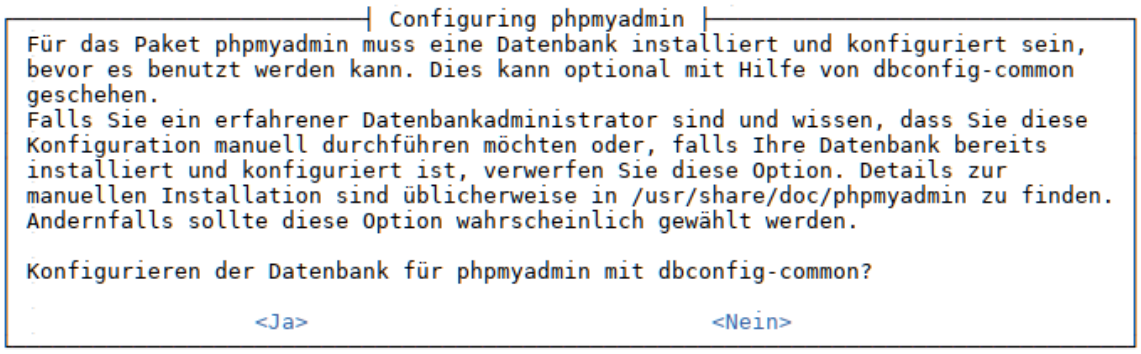


Abbildung 41: PHPMyAdmin Datenbank

In den folgenden Dialogen benötigen wir das Passwort für den Datenbankbenutzer *phpmyadmin* (siehe das Kapitel *Vorbereitungen*).

5.6.5 Weitere Software

Wer sich mit der Entwicklung von Webseiten befasst, kann ein CMS zum Beispiel, WordPress, Drupal oder Joomla installieren, sollte zuvor jedoch unsere Handbuchseiten LAMP-Apache und LAMP-MariaDB für die Konfiguration des Servers und MariaDB berücksichtigen.

5.6.6 Statusaugaben Log-Dateien

Apache

Der Konfigurationsstatus des Apache Webservers wird mit `apache2ctl -S` ausgegeben.

Die Ausgabe zeigt den Status ohne Änderungen an der Konfiguration unmittelbar nach der Installation.

```
# apache2ctl -S
  AH00558: apache2: Could not reliably determine the server's
  fully qualified domain name, using 127.0.1.1. Set the '↙
     ServerName'
  directive globally to suppress this message
  VirtualHost configuration:
  [::1]:80               127.0.0.1 (/etc/apache2/sites-↙
     enabled/000-default.conf:1)
  127.0.0.1:80           127.0.0.1 (/etc/apache2/sites-↙
     enabled/000-default.conf:1)
  ServerRoot: "/etc/apache2"
  Main DocumentRoot: "/var/www/html"
  Main ErrorLog: "/var/log/apache2/error.log"
  Mutex default: dir="/var/run/apache2/" mechanism=default
  Mutex mpm-accept: using_defaults
  Mutex watchdog-callback: using_defaults
  PidFile: "/var/run/apache2/apache2.pid"
  Define: DUMP_VHOSTS
  Define: DUMP_RUN_CFG
  User: name="www-data" id=33
  Group: name="www-data" id=33
```

Die Handbuchseite LAMP-Apache enthält eine Reihe von Hinweisen zur Anpassung der Konfiguration.
Das Verzeichnis `/var/log/apache2/` enthält die Log-Dateien. Ein Blick in diese ist behilflich um Fehlerursachen zu erkennen.

MariaDB

In der Konsole zeigt der Befehl

```
# systemctl status mariadb.service
```

den aktuellen Status von MariaDB und die letzten zehn Logeinträge.
Die letzten zwanzig Zeilen des Systemd-Journals zeigt der Befehl

```
# journctl -n 20 -u mariadb.service
```

und

```
# journctl -f -u mariadb.service
```

hält die Verbindung zum Journal offen und zeigt laufend die neuen Einträge.
Weitere Informationen liefert die Handbuchseite LAMP-MariaDB

PHP

Die Fehlermeldungen von PHP speichert der Apache Server in seinen Log-Dateien unter `/var/log/apache2/`. Fehlerhafte PHP-Funktionen erzeugen eine Meldung in der aufgerufenen Webseite.
Dieses Verhalten lässt sich in den php.ini-Dateien des jeweiligen Interface konfigurieren.
Siehe die Handbuchseite LAMP-PHP

5.6.7 Troubleshooting

Die hier aufgeführten Beispiele zeigen exemplarisch einige Möglichkeiten der Fehlersuche.

Dateirecht in “DocumentRoot”

Sollte unmittelbar nach der Installation der Aufruf der Dateien `index.html` und `info.php` fehlschlagen, bitte unbedingt zuerst die Eigentümer- und Gruppenzugehörigkeit des Webseitenverzeichnisses überprüfen und ggf. ändern:

```
# ls -la /var/www/html
drwxr-xr-x 2 www-data www-data  4096 14. Dez 18:56 .
drwxr-xr-x 3 root     root      4096 14. Dez 18:30 ..
-rw-r--r-- 1 www-data www-data 10701 14. Dez 19:04 index.html
-rw-r--r-- 1 root     root        20 14. Dez 19:32 info.php
```

In diesem Fall wird die Apache Testseite angezeigt, die PHP-Statusseite nicht. Dann hilft ein beherztes

```
# chown -R www-data:www-data /var/www/html
```

Nun sollten sich beide Seiten aufrufen lassen.

HTML-Seiten-Ladefehler

Die Webseite http://localhost/index.html wird nicht angezeigt und der Browser meldet einen Seiten-Ladefehler.

Wir fragen den Status des Apache Webservers ab:

```
# systemctl status apache2.service●
 apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; ⬋
        enabled; vendor preset: enabled)
     Active: failed (Result: exit-code) since Mon 2020-12-14 ⬋
        18:29:23 CET; 13min ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 4420 ExecStart=/usr/sbin/apachectl start (code=⬋
       exited, status=1/FAILURE)

Dez 14 18:29:23 lap1 systemd[1]: Starting The Apache HTTP ⬋
   Server...
Dez 14 18:29:23 lap1 apachectl[4423]: AH00526: Syntax error ⬋
   on line 63 of /etc/apache2/conf-enabled/security.conf:
[...]
```

Wir sehen, dass die Datei *"security.conf"* in Zeile 63 einen Fehler aufweist.
Wir bearbeiten die Datei und versuchen es noch einmal.

```
# systemctl start apache2.service
# systemctl status apache2.service●
 apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; ⬋
        enabled; vendor preset: enabled)
     Active: active (running) since Mon 2020-12-14 18:34:59 ⬋
        CET; 3s ago
[...]
```

Generell ist nach jeder Änderung der Konfiguration ein Reload oder Restart des Apache notwendig.

Apache Log-Dateien prüfen

Ein Blick in die Logdateien unter `/var/log/apache2/` hilft um Fehler in der Konfiguration des Netzwerks oder des Apache Servers zu erkennen.

PHP, info.php nur weiße Seite

Das bedeutet, dass PHP aktiv ist, aber die Seite nicht anzeigen kann.
Bitte überprüfen:

- Der Inhalt der Datei `info.php` muss exakt dem im Kapitel PHP gegebenem Beispiel entsprechen.
- Die Dateirechte, wie zu Beginn des Kapitels Troubleshooting erläutert, prüfen und ggf. ändern.
- Zusätzliche PHP-Module wurden installiert oder die Konfiguration geändert und der Webserver nicht neu gestartet.
  Dann hilft:

```
# systemctl restart apache2.service
```

phpMyAdmin - Error

Der Aufruf von http://localhost/phpmyadmin schlägt mit der Meldung *"phpMyAdmin - Error"* fehl und die folgenden Informationen werden angezeigt.

```
Error during session start; please check your PHP and/or ↙
   webserver log file and
configure your PHP installation properly. Also ensure that ↙
   cookies are enabled
in your browser.

session_start(): open(SESSION_FILE, O_RDWR) failed: ↙
   Permission denied (13)
session_start(): Failed to read session data: files (path: /↙
   var/lib/php/sessions)
```

Die Berechtigungen für den Ordner `/var/lib/php/sessions` prüfen:

```
# ls -l /var/lib/php/
```

Die Ausgabe sollte diese Zeile enthalten:

```
drwx-wx-wt 2 root root 4096 14. Dez 17:32 sessions
```

Zu beachten ist das Sticky-Bit (**t**) und der Eigentümer **root.root**. Bei Abweichungen beheben wir den Fehler.

```
# chmod 1733 /var/lib/php/sessions
# chown root:root /var/lib/php/sessions
```

Nun ist der Login zu phpmyadmin möglich.

5.6.7.1 Wenn nichts hilft Die Installation des LAMP-Stack ist in weniger als fünfzehn Minuten erledigt. Eine Fehlersuche kann jedoch Stunden in Anspruch nehmen.
Deshalb ist es, sofern die zuvor genannten Maßnahmen zu keiner Lösung führen, sinnvoll den LAMP-Stack oder Teile davon zu entfernen und neu zu installieren. Wenn, wie im Kapitel Vorbereitungen erwähnt, apt aufgeräumt wurde, hilft der Befehl `apt purge` um die zuvor installierten Pakete mit ihren Konfigurationsdateien zu entfernen ohne das irgendwelche anderen Pakete stören.

Hier ein Beispiel mit Apache:

```
# apt purge apache2
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut.
Statusinformationen werden eingelesen.... Fertig
Die folgenden Pakete wurden automatisch installiert und ↙
   werden nicht mehr benötigt:
apache2-data apache2-utils
Verwenden Sie »apt autoremove«, um sie zu entfernen.
Die folgenden Pakete werden ENTFERNT:
  apache2*
0 aktualisiert, 0 neu installiert, 1 zu entfernen und 0 nicht↙
    aktualisiert.
```

“Apache2” wird entfernt und die Pakete *“apache2-data”* und *“apache2-utils”* bleiben noch erhalten.
Jetzt bitte **nicht** *“apt autoremove”* verwenden, denn dann bleiben die Konfigurationsdateien, in denen möglicherweise der Fehler liegt, zurück.
Wir verwenden den Befehl `apt purge`.

```
# apt purge apache2-data apache2-utils
```

Bei Bedarf verfahren wir mit den anderen Programmteilen ebenso. Anschließend starten wir einen neuen Versuch.

5.6.8 Sicherheit

Die bis hierher erklärte Installation führt zu einem Webserver der **“offen wie ein Scheunentor für Jedermann ist”**. Deshalb sollte er ausschließlich autark an einem Arbeitsplatz verwendet und nicht mit dem privaten Netzwerk und auf keinen Fall mit dem Internet verbunden werden.

Für die Absicherung des Servers bitte die Handbuchseiten

LAMP-Apache LAMP-MariaDB LAMP-PHP

bezüglich der Konfiguration beachten.

Danach kann der Server, ausschließlich für System- und Software- Aktualisierungen, temporär über eine zweite Netzwerkschnittstelle mit dem Internet verbunden werden.

Zuletzt bearbeitet: 2021-11-29

5.7 Apache einrichten

Diese Handbuchseite basiert auf Apache 2.4.46.

Unserem Beispiel aus der Installationsanleitung entsprechend, wollen wir einen *“LAMP-Testserver für Entwickler”* aufsetzen, der über LAN direkt mit dem Arbeitsplatz-PC verbunden ist. Darüber hinaus soll es aus Gründen der Sicherheit für den Server keine Verbindung zu einem lokalen Netzwerk oder gar zum Internet geben.
Einzige Ausnahme: Der Server wird temporär und ausschließlich für System- und Software- Aktualisierungen über eine zweite Netzwerkschnittstelle mit dem Internet verbunden.

5.7.1 Apache im Dateisystem

Debian hat die Dateien des Apache entsprechend ihrer Funktion vollständig in das Dateisystem integriert.

- In `/usr/sbin/` das ausführbare Programm `apache2`.
- In `/usr/lib/apache2/modules/` die installierten Module für Apache.
- In `/usr/share/apache2/` Dateien, die auch für andere Programme verfügbar sind.
- In `/etc/apache2/` die Konfigurationsverzeichnisse und -dateien.
- In `/var/www/html/` die vom Benutzer angelegte Webseite.
- In `/run/apache2/` und `/run/lock/apache2/` zur Laufzeit notwendige Systemdateien.
- In `/var/log/apache2/` verschiedene Log-Dateien.

Wichtig ist die Unterscheidung zwischen den verwendeten Variablen **ServerRoot** und **DocumentRoot**.

“ServerRoot” ist das Konfigurationsverzeichnis, also `/etc/apache2/`.
“DocumentRoot” beinhaltet die Webseitendaten, also `/var/www/html/`.

5.7.2 Verbindung zum Server

Die Verbindung zwischen Testserver und PC wird in das IPv4-Netzwerksegment **192.168.3.xxx** gelegt, während die Internetverbindung des PC außerhalb dieses Netzwerksegmentes erfolgt. Die verwendeten Daten sind:

Server
IP: 192.168.3.1/24
Name: server1.org
Alias: www.server1.org

PC
IP: 192.168.3.10/24
Name: pc1

Wir legen von der Datei `/etc/hosts` auf dem Server und auf dem PC eine Sicherungskopie an und fügen beiden die notwendigen Zeilen hinzu.

- Server *"/etc/hosts"*:

```
cp /etc/hosts /etc/hosts_$(date +%f)
echo "192.168.3.1 server1.org   www.server1.org" >> /etc/↙
   hosts
echo "192.168.3.10 pc1" >> /etc/hosts
```

- PC *"/etc/hosts"*:

```
cp /etc/hosts /etc/hosts_$(date +%f)
echo "192.168.3.1 server1.org   www.server1.org" >> /etc/↙
   hosts
```

Als nächstes geben wir im `NetworkManager` die Daten für den Server in die rot umrandeten Feldern ein. Die Methode wird von *"Automatisch (DHCP)"* auf *"Manuell"* geändert und in die Adressfelder tragen wir die zu Beginn genannten Werte ein.

Zusätzlich sollte im Reiter *"Allgemein"* die Option *"Automatisch mit Priorität verbinden"* aktiviert sein.
Sinngemäß nehmen wir am PC die entsprechenden Einstellungen für die verwendete LAN-Schnittstelle vor.

Am PC testen wir die Verbindung in der Konsole mit

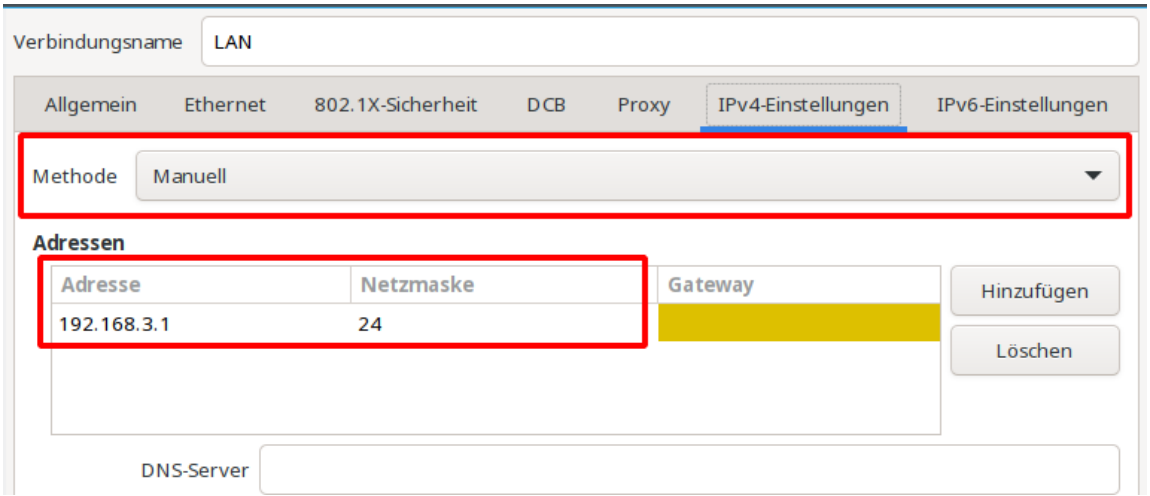


Abbildung 42: Server - Dateneingabe im NetworkManager

```
$ ping -c3 www.server1.org
```

und bei Erfolg prüfen wir gleich die Funktion von Apache, indem wir in die Adresszeile des Webbrowsers http://www.server1.org/index.html eingeben.

Die Apache-Begrüßungsseite mit *"It works!"* sollte erscheinen.

5.7.3 Apache Konfiguration

Die Konfigurationsdateien und -verzeichnisse befindet sich im *"ServerRoot"* Verzeichnis `/etc/apache2/`.
Die zentrale Konfigurationsdatei ist `apache2.conf`. Sie wird in der Regel nicht bearbeitet, da viele Konfigurationen in separaten Dateien vorliegen. Die Aktivierung und Deaktivierung erfolgt über Sym-Links. Das hat den Vorteil, dass eine Reihe verschiedener Konfigurationen vorhanden sind und nur die benötigten eingebunden werden.

Bei den Konfigurationsdateien handelt es sich um Textdateien, welche mit einem Editor und Root-Rechten angelegt bzw. editiert werden. Der Name der Datei darf beliebig sein, aber die Dateiendung muss *".conf"* lauten. Die gültigen Direktiven, die in den Konfigurationsdateien verwendet werden dürfen, beschreibt die Apache Dokumentation ausführlich.

Die Dateien liegen in den Verzeichnissen

`/etc/apache2/conf-available`,
`/etc/apache2/mods-available` und
`/etc/apache2/sites-available`.

Ihre Aktivierungs-Links finden wir in

`/etc/apache2/conf-enable`,
`/etc/apache2/mods-enable` und
`/etc/apache2/sites-enable`.

Um eine .conf-Datei zu aktivieren bzw. deaktivieren benutzen wir die Befehle `a2enconf` und `a2disconf`. Das erstellt oder entfernt die Aktivierungs-Links.

```
a2enconf NAME_DER_DATEI.conf
```

Aktiviert die Konfiguration. Die Deaktivierung erfolgt entsprechend mit:

```
a2disconf NAME_DER_DATEI.conf
```

In gleicher Weise verfahren wir bei Modulen und Virtual-Hosts mit den Befehlen `a2enmod`, `a2ensite` und `a2dismod`, `a2dissite`.

Der Apache Webserver liest mit dem Befehl

```
systemctl reload apache2.service
```

die geänderte Konfiguration ein.

Nun kommen wir wieder auf unseren *“LAMP-Testserver für Entwickler”* zurück und passen die Konfiguration an die Serverdaten an.

1. Datei `/etc/apache2/apache2.conf`

   Es ist eine der wenigen Ausnahmen die `apache2.conf` zu editieren. Wir fügen zu Beginn des Abschnitts *“Global configuration”* die folgende Zeile ein:

   ```
   ServerName 192.168.3.1
   ```

   Hiermit teilen wir dem Apache-Webserver die IP-Adresse mit, unter der das Entwicklungsprojekt erreichbar sein soll und unterdrücken Umleitungen zur IP 127.0.1.1 mit Fehlermeldungen.

2. Neue `sites`-Datei

   Mit dem Texteditor unserer Wahl erstellen wir die Datei `/etc/apache2/sites-available/server1.conf` z. B.

```
mcedit /etc/apache2/sites-available/server1.conf
```

   und fügen den folgenden Inhalt ein, speichern die Datei und beenden den Editor.

```
<VirtualHost *:80>
    ServerName server1.org
    ServerAlias www.server1.org
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error_server1.log
    CustomLog ${APACHE_LOG_DIR}/access_server1.log ↙
       combined
</VirtualHost>
```

   Anschließend stellen wir die Konfiguration auf den neuen *“VirtualHost”* um und geben die Änderungen dem Apache Webserver bekannt.

```
# a2ensite server1.conf
  Enabling site server1.
[...]

# a2dissite 000-default.conf
  Site 000-default disabled.
[...]

systemctl reload apache2.service
```

5.7.4 Benutzer und Rechte

Der Apache Webserver läuft mit <user> und <group> **www-data.www-data** und *“DocumentRoot”* gehört unmittelbar nach der Installation **root.root**.
Um Benutzern Schreibrechte für die in *“DocumentRoot”* enthaltenen Dateien zu gegeben, sollte dafür eine neue Gruppe angelegt werden. Es ist nicht sinnvoll die

bestehende Gruppe *“www-data”* zu nutzten, da mit den Rechten dieser Gruppe Apache läuft.
Wir nennen die neue Gruppe `developer`.

Mit CMS

Wird ein Content-Management-System (Software zur gemeinschaftlichen Bearbeitung von Webseiten-Inhalten) hinzugefügt, bereiten wir *“DocumentRoot”* entsprechend vor:

1. Gruppe anlegen und dem Benutzer zuweisen.

   ```
   groupadd developer
   adduser <user> developer
   chgrp developer /var/www/html
   ```

   Um die neuen Rechte zu aktivieren, muss man sich einmal ab- und neu anmelden oder als Benutzer den Befehl newgrp verwenden.

   ```
   $ newgrp developer
   ```

2. SGID-Bit für *“DocumentRoot”* setzen,
   damit alle hinzukommenden Verzeichnisse und Dateien die Gruppe `developer` erben.

   ```
   chmod g+s /var/www/html
   ```

3. Eigentümer und Dateirechte anpassen,
   damit Unbefugte keinen Zugriff erhalten und der Apache Webserver einwandfrei läuft.
   Wir schauen uns die derzeitigen Rechte an:

   ```
   # ls -la /var/www/html
   insgesamt 24
   drwxr-sr-x 2 root developer  4096  9. Jan 19:32 . ↙
                (DocumentRoot mit SGID-Bit)
   drwxr-xr-x 3 root root       4096  9. Jan 19:04 .. ↙
               (Das übergeordnete Verzeichnis /var/www)
   -rw-r--r-- 1 root developer 10701  9. Jan 19:04 index.↙
      html
   -rw-r--r-- 1 root developer    20  9. Jan 19:32 info.php
   ```

Wir ändern für *“DocumentRoot”* den Eigentümer zu *“www-data”*, geben der Gruppe Schreibrecht und entziehen allen anderen auch das Leserecht. Alles rekursiv.

```
chown -R www-data /var/www/html
chmod -R g+w /var/www/html
chmod -R o-r /var/www/html
```

Das Ergebnis überprüfen wir noch einmal.

```
# ls -la /var/www/html
insgesamt 24
dr-xrws--x 2 www-data developer  4096  9. Jan 19:32 .
drwxr-xr-x 3 root     root       4096  9. Jan 19:04 ..
-rw-rw---- 1 www-data developer 10701  9. Jan 19:04 index⇙
   .html
-rw-rw---- 1 www-data developer    20  9. Jan 19:32 info.⇙
   php
```

Jetzt haben in *“DocumentRoot”* nur Mitglieder der Gruppe `developer`⇙ Schreibrecht, der Apache Webserver kann die Dateien lesen und schreiben, allen anderen wird der Zugriff verweigert.

4. Nachteile dieser Einstellungen

   Beim Anlegen neuer Verzeichnisse und Dateien unterhalb *“DocumentRoot”* ist der Eigentümer der jeweilige `User` und nicht `www-data`. Dadurch kann der Apache-Webserver die Dateien nicht lesen.
   Abhilfe schafft eine `Systemd Path Unit`, die Änderungen unterhalb *“DocumentRoot”* überwacht und die Eigentümer- und Dateirechte anpasst. (Siehe das Beispiel in der Handbuchseite Systemd-Path.)

Ohne CMS

Bei statischen Webseiten ist ein Content-Management-System vielfach nicht notwendig und bedeutet nur ein weiteres Sicherheitsrisiko und erhöhten Wartungsaufwand. Zusätzlich zu den zuvor getätigten Einstellungen kann dem Apache-Webserver das Schreibrecht an *“DocumentRoot”* entzogen werden, um die Sicherheit zu stärken, denn für den Fall, dass ein Angreifer eine Lücke in Apache findet, erhält er dadurch keine Schreibrechte in *“DocumentRoot”*.

```
chmod -R u-w /var/www/html
```

5.7.5 Sicherheit - Apache Standard

Wichtige Absicherungen enthält die Datei `/etc/apache2/apache2.conf` bereits standardmäßig.

Die nachfolgenden drei Direktiven verhindern den Zugang zum root-Dateisystem und geben dann die beiden vom Apache-Webserver verwendeten Verzeichnisse `/usr/share` und `/var/www` frei.

```
<Directory />
  Options FollowSymLinks
  AllowOverride None
  Require all denied
</Directory>

<Directory /usr/share>
  AllowOverride None
  Require all granted
</Directory>

<Directory /var/www/>
  Options Indexes FollowSymLinks
  AllowOverride None
  Require all granted
</Directory>
```

Die Optionen *“FollowSymLinks”* und *“Indexes”* bergen ein Sicherheitsrisiko und sollten geändert werden, sofern sie nicht unbedingt notwendig sind. Siehe weiter unten.

Die folgende Direktive unterbindet die Anzeige der Dateien `.htaccess` und `.htpasswd`.

```
<FilesMatch "^\.ht">
  Require all denied
</FilesMatch>
```

5.7.6 Sicherheit - weitere Konfigurationen

- In der Datei `/etc/apache2/apache2.conf`

  "FollowSymLinks" kann dazu führen, dass Inhalte außerhalb *"DocumentRoot"* gelistet werden.
  "Indexes" listet den Inhalt eines Verzeichnisses, sofern keine `index.html` oder `index.php` usw. vorhanden ist.

  Es ist empfehlenswert *"FollowSymLinks"* zu entfernen und die Projektdaten alle unterhalb *"DocumentRoot"* abzulegen. Für die Option *"Indexes"* ist der Eintrag zu ändern in

  ```
  Options -Indexes
  ```

  wenn die Anzeige des Verzeichnisinhaltes nicht erwünscht ist.
  Alternativ erstellt man in dem Verzeichnis eine leere index-Datei, die an Stelle des Verzeichnisinhaltes an den Client ausgeliefert wird. Zum Beispiel für das upload-Verzeichnis:

  ```
  $ echo "<!DOCTYPE html>" > /var/www/html/upload/index.↙
     html
       oder
  $ echo "<?php" > /var/www/html/upload/index.php
  ```

- In der Host-Konfiguration `/etc/apache2/sites-available/server1.↙conf`

  können wir mit dem `<Directory>`-Block alle IP-Adressen sperren, außer die darin gelisteten.

  ```
  <Directory "/var/www/html">
      Order deny,allow
      Deny from all
      Allow from 192.168.3.10
      Allow from 192.168.3.1
  </Directory>
  ```

- **"merging"** der Konfiguration

Die Direktiven der Konfiguration verteilen sich auf eine ganze Reihe von Dateien innerhalb *“ServerRoot”* und auf die *“.htaccess”*-Dateien in *“DocumentRoot”*. Es ist deshalb besonders wichtig zu wissen an welcher Stelle die Direktive zu platzieren ist, um die gewünschte Wirkung zu erzielen.
Wir empfehlen dringend die Webseite
apache.org - How the sections are merged
intensiv zu Rate zu ziehen.

- Der **Eigentümer** von *“DocumentRoot”*

  ist nach der Installation **root.root** und sollte unbedingt geändert werden. Siehe hierzu das Kapitel Benutzer und Rechte.

5.7.7 HTTPS verwenden

Ohne HTTPS geht heute kein Webseitenprojekt an den Start.
Wie man ein Zertifikat erlangt beschreibt die Webseite HTTP-Guide ausführlich und leicht verständlich.

Wir legen zuerst die nötigen Ordner innerhalb *“DocumentRoot”* an:

```
cd /etc/apache2/
/etc/apache2/# mkdir ssl ssl/certs ssl/privat
```

In diesen legen wir die Certifikatsdatei `server1.org.crt` und den privaten Schlüssel `server1.org.key` ab.

Dann sichern wir die Verzeichnisse gegen unbefugten Zugriff.

```
/etc/apache2/# chown -R root.root ssl
/etc/apache2/# chmod -R o-rwx ssl
/etc/apache2/# chmod -R g-rwx ssl
/etc/apache2/# chmod u-w ssl/certs/server1.org.crt
/etc/apache2/# chmod u-w ssl/private/server1.org.key
```

Der ls-Befehl zur Kontrolle:

```
/etc/apache2/# ls -la ssl
   insgesamt 20
   drwx------ 5 root root 4096 25. Jan 18:17 .
   drwxr-xr-x 9 root root 4096 25. Jan 18:43 ..
```

```
   drwx------ 2 root root 4096 25. Jan 18:16 certs
   drwx------ 2 root root 4096 25. Jan 18:16 private

/etc/apache2/# ls -l ssl/certs
   -r-------- 1 root root 1216 25. Jan 15:27 server1.org.crt
```

5.7.8 Sicherheits Tipps

- Die Apache Dokumentation enhält eine empfehlenswerte Seite mit diversen Tipps zur Absicherung.
  apache.org - Security Tipps (englisch)
- Darüber hinaus finden sich im Internet zahlreiche Hinweise zum sicheren Betrieb des Apache Webservers.
- Die regelmäßige Kontrolle der Logdateien in `/var/log/apache2/` hilft um Fehler oder Sicherheitslücken zu erkennen.
- Sollte der Server, anders als in dieser Handbuchseite vorgesehen, mit dem lokalen Netzwerk oder mit dem Internet verbunden werden, ist eine Firewall unerlässlich.

5.7.9 Integration in Apache2

Das ssl-Modul ist in Apache per default aktviert. Es genügt die Datei `/etc/apache2/sites-available/server1.conf` zu bearbeiten.

- Eine neue VirtualHost-Directive wird zu Beginn eingefügt. Diese leitet eingehende Client-Anfragen von Port 80 mittels *“Redirect”* auf Port 443 (ssl) weiter.
- Die bisherige VirtualHost-Directive wird auf Port 443 umgeschrieben.
- Nach den Standard Host-Anweisungen fügen wir die SSL-Anweisungen ein.
- Für den Fall, dass unser Webprojekt dynamisch generierte Webseiten enthalten soll, werden die beiden letzten FileMatch- und Directory-Direktiven mit der “*SSLOptions*”-Anweisung eingefügt.

Die erweiterte `server1.conf` weist dann folgenden Inhalt auf:

```
<VirtualHost *:80>
    ServerName server1.org
    ServerAlias www.server1.org
    Redirect / https://server1.org/
</VirtualHost>

<VirtualHost *:443>
    ServerName server1.org
    ServerAlias www.server1.org
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error_server1.log
    CustomLog ${APACHE_LOG_DIR}/access_server1.log combined

    SSLEngine on
    SSLProtocol all -SSLv2 -SSLv3
    SSLCertificateFile      /etc/apache2/ssl/certs/server1.↙
       org.crt
    SSLCertificateKeyFile /etc/apache2/ssl/private/server1.↙
       org.key

    <Directory "/var/www/html">
      Order deny,allow
      Deny from all
      Allow from 192.168.3.10
      Allow from 192.168.3.1
    </Directory>

    <FilesMatch "\.(cgi|shtml|phtml|php)$">
      SSLOptions +StdEnvVars
    </FilesMatch>

    <Directory /usr/lib/cgi-bin>
      SSLOptions +StdEnvVars
    </Directory>
</VirtualHost>
```

Für den Fall, dass unser fertiges Projekt später bei einem Hoster ohne Zugriff auf *"ServerRoot"* liegt (das ist die Regel), können wir in *"DocumentRoot"* die Datei `.htaccess` um eine Rewrite-Anweisung ergänzen bzw. die Datei mit der Rewrite-Anweisung anlegen.

```
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteCond %{HTTPS} !=on
RewriteRule ^ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
</IfModule>
```

5.7.10 Quellen Apache

apache.org - Dokumentation (teilweise deutsch)
apache.org - Konfigurationsdateien
apache.org - SSL Howto
HTTPS Guide - Servercertifikate erstellen und integrieren

Zuletzt bearbeitet: 2021-07-12

5.8 MariaDB einrichten

5.8.1 MariaDB im Dateisystem

Debian hat die Dateien von MariaDB entsprechend ihrer Funktion vollständig in das Dateisystem integriert.

- In `/usr/bin/` das ausführbare Programm *“mariadb”*
  - und der Link `mysql`, der auf `/usr/bin/mariadb` verweist.
- In `/usr/lib/mysql/plugin/` die installierten Plugin für MariaDB.
- In `/usr/share/mysql/` Gemeinsam genutzte Programmteile und Lokalisierungen.
- In `/etc/mysql/` die Konfigurationsverzeichnisse und -dateien.
- In `/var/lib/mysql/` die Datenbanken und Log-Dateien.
- In `/run/mysqld/` zur Laufzeit notwendige Systemdateien.

Innerhalb der zuvor genannten Verzeichnisse sollten die Dateien tunlichst nicht manuell bearbeitet werden. Einzige Ausnahme ist die Konfiguration von MariaDB unterhalb `/etc/mysql/`, sofern man genau weiß wie vorzugehen ist. Anderen Falls benutzt man das MariaDB-CLI oder ein Frontend wie phpMyAdmin.

5.8.2 Erstkonfiguration

Nach der Installation, wie sie in LAMP-Testserver für Entwickler beschrieben wurde, ist MariaDB *‘offen wie ein Scheunentor für jedermann’*, denn in der Grundeinstellung werden die beiden Benutzer **root** und **anonymous**, ohne Passwort erstellt und eine Testdatenbank angelegt.

Deshalb rufen wir das Programm **`mysql_secure_installation`** im Root-Terminal auf.
Hier nehmen wir eine ganze Reihe von Einstellungen zur Absicherung der Datenbank vor. Die notwendigen Eingaben sind so gekennzeichnet: `«--[ ]`.

```
# mysql_secure_installation
```

```
In order to log into MariaDB to secure it, we'll need the ⬋
   current
password for the root user. If you've just installed MariaDB,⬋
    and
you haven't set the root password yet, the password will be ⬋
   blank,
so you should just press enter here.

Enter current password for root (enter for none): «--[Enter]
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures ⬋
   that nobody can
log into the MariaDB root user without the proper ⬋
   authorisation.

You already have your root account protected, so you can ⬋
   safely answer 'n'.

Switch to unix_socket authentication [Y/n]:   «--[n]
 ... skipping.

You already have your root account protected, so you can ⬋
   safely answer 'n'.

Change the root password? [Y/n]:   «--[y]
New password:                      «--[⬋
   mein_mariadb_root_passwort]
Re-enter new password:             «--[⬋
   mein_mariadb_root_passwort]
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MariaDB installation has an anonymous user, ⬋
   allowing anyone
to log into MariaDB without having to have a user account ⬋
   created for
```

```
them. This is intended only for testing, and to make the ↙
   installation
go a bit smoother. You should remove them before moving into ↙
   a
production environment.

Remove anonymous users? [Y/n]:   «--[y]
... Success!

Normally, root should only be allowed to connect from '↙
   localhost'. This
ensures that someone cannot guess at the root password from ↙
   the network.

Disallow root login remotely? [Y/n]   «--[y]
... Success!

By default, MariaDB comes with a database named 'test' that ↙
   anyone can
access. This is also intended only for testing, and should be↙
    removed
before moving into a production environment.

Remove test database and access to it? [Y/n]   «--[y]
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes ↙
   made so far
will take effect immediately.

Reload privilege tables now? [Y/n]   «--[y]
... Success!

Cleaning up...
```

```
All done! If you've completed all of the above steps, your ↙
   MariaDB
installation should now be secure.

Thanks for using MariaDB!
```

Im Ergebnis hat der Benutzer **root** ein (hoffentlich sicheres) Passwort erhalten und er kann sich nicht mehr remote einloggen. Der Benutzer **anonymous** und die Datenbank *"Test"* wurden entfernt.

5.8.3 MariaDB CLI

Das Commandline Interface erreichen wir im Terminal durch die Eingabe von **`mariadb -u \<user\> -p`**. Nach der Eingabe des Passwortes sehen wir die Begrüßung und den neuen Promt `MariaDB [(none)]>`.

```
# mariadb -u root -p
Enter password:
Welcome to the MariaDB monitor.  [...]

MariaDB [(none)]>
```

Aus Sicherheitsgründen loggen wir uns nur zu Beginn als Benutzer **root** ein, um die Projektdatenbank, einen Benutzer für die alltäglichen Arbeiten an dieser und einen Benutzer als Ersatz für root anzulegen.
Später im Abschnitt phpMyAdmin entziehen wir dem Benutzer root die allumfassenden Rechte, damit ein potentieller Angreifer an dieser Stelle erfolglos bleibt.

5.8.3.1 Eine Datenbank anlegen

Wir sind noch im Terminal angemeldet und erstellen für unser Projekt eine neue Datenbank:

```
MariaDB [(none)]> CREATE DATABASE sidu;
Query OK, 1 row affected (0.002 sec)
```

Das ist schon alles. Falls wir diese Datenbank löschen wollen lautet der Befehl `DROP DATABASE sidu;`

5.8.3.2 Einen Benutzer anlegen Zuerst erstellen wir unseren Projekt-Benutzer mit dem Namen **tomtom** und weisen ihm ausschließlich alle Rechte an der Projekt-Datenbank *“sidu”* zu:

```
MariaDB [(none)]> CREATE USER tomtom@localhost IDENTIFIED BY ⇙
   '<hier ein Passwort für tomtom eingeben>';
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]> GRANT ALL ON sidu.* TO tomtom@localhost;
Query OK, 0 rows affected (0.001 sec)
```

Nun die gleiche Prozedur für den Benutzer **chef**, der die Aufgabe von root übernehmen soll.

```
MariaDB [(none)]> CREATE USER chef@localhost IDENTIFIED BY '<⇙
   hier ein Passwort für chef eingeben>';
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]> GRANT ALL ON *.* TO chef@localhost WITH ⇙
   GRANT OPTION;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
```

Die neuen Benutzer unterscheiden sich in ihren Rechten.

“tomtom” hat alle Rechte nur für die Datenbank *“sidu”* (sidu.*).
“chef” hat alle Rechte an allen Datenbanken (*.*) und Benutzern (WITH GRANT OPTION).

Der Benutzer **chef** kann somit die Funktion des Benutzers **root** übernehmen und den Benutzer **tomtom** verwenden wir für Arbeiten an unserer Projektdatenbank. Den Logout erledigt: `\q`.

```
MariaDB [(none)]> \q
Bey
#
```

5.8.3.3 Abfragen Wir schauen uns das Ergebnis in Terminal an, diesmal als Benutzer *“chef”*.
Zuerst die Benutzer und dann die vorhandenen Datenbanken.

```
MariaDB [(none)]> SELECT User,Host FROM mysql.user;
+-------------+-----------+
| User        | Host      |
+-------------+-----------+
| chef        | localhost |
| mariadb.sys | localhost |
| mysql       | localhost |
| phpmyadmin  | localhost |
| root        | localhost |
| tomtom      | localhost |
+-------------+-----------+
6 rows in set (0.002 sec)

MariaDB [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| phpmyadmin         |
| sidu               |
+--------------------+
5 rows in set (0.001 sec)
```

Wenn wir uns von MariaDB abmelden und als Benutzer *“tomtom”* wieder anmelden, sehen die beiden Abfragen wie folgt aus:

```
MariaDB [(none)]> SELECT User,Host FROM mysql.user;
ERROR 1142 (42000): SELECT command denied to user 'tomtom'@'⇙
   localhost' for table 'user'

MariaDB [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
```

```
+--------------------+
| information_schema |
| sidu               |
+--------------------+
2 rows in set (0.001 sec)
```

Es ist gut zu erkennen, dass der Benutzer *“tomtom”* keinen Zugriff auf systemrelevante Daten erhält.

5.8.4 phpMyAdmin

Wie zuvor gesehen, lässt sich MariaDB vollständig über die Komandozeile verwalten. Wer die Syntax beherrscht, und dafür ist profundes Fachwissen erforderlich, kommt auf diesem Weg schnell zum gewünschten Ergebnis.

Wir verwenden das für weniger erfahrene Benutzer besser geeignete Progrann `phpMyAdmin` und geben in die Adresszeile des Browsers
http://localhost/phpmyadmin/
ein. Sollten wir die Konfiguration entsprechend der Handbuchseite LAMP - Apache bereits durchlaufen haben, lautet der Aufruf
https://server1.org/phpmyadmin/

Um, wie oben angeführt, dem Datenbank-Admin **root** die Rechte zu entziehen, benutzen wir im Anmeldefenster gleich unseren neuen Datenbank-Admin **chef** mit seinem Passwort.

Im Startfenster sehen wir in der linken Spalte alle Datenbanken. Im Hauptteil wählen wir den Reiter **`Benutzerkonten`**.

Die Benutzerkontenübersicht stellt alle Benutzer und in Kurzform deren Rechte dar. Wir wählen hier für den Benutzer **root** den Schalter **`Rechte ändern`**.

Nun sehen wir für den Benutzer **root** die detaillierten Rechte. Hier entziehen wir ihm erst einmal alle Rechte (1a), erteilen dann im Bereich `Administration` das Recht *“Super”* (1b) und führen die Aktion aus, indem wir ganz unten rechts auf dieser Seite den `OK`-Button anklicken (im Screenshot nicht sichtbar).

Anschließend gehen wir über den `Datenbank`-Schalter (2) zur nächsten Seite.

Nach Auswahl der Datenbank *“mysql”* und `OK` öffnet sich diesmal ein Fenster mit den detaillierten Rechten an der Datenbank *“mysql”* für den Benutzer **root**.

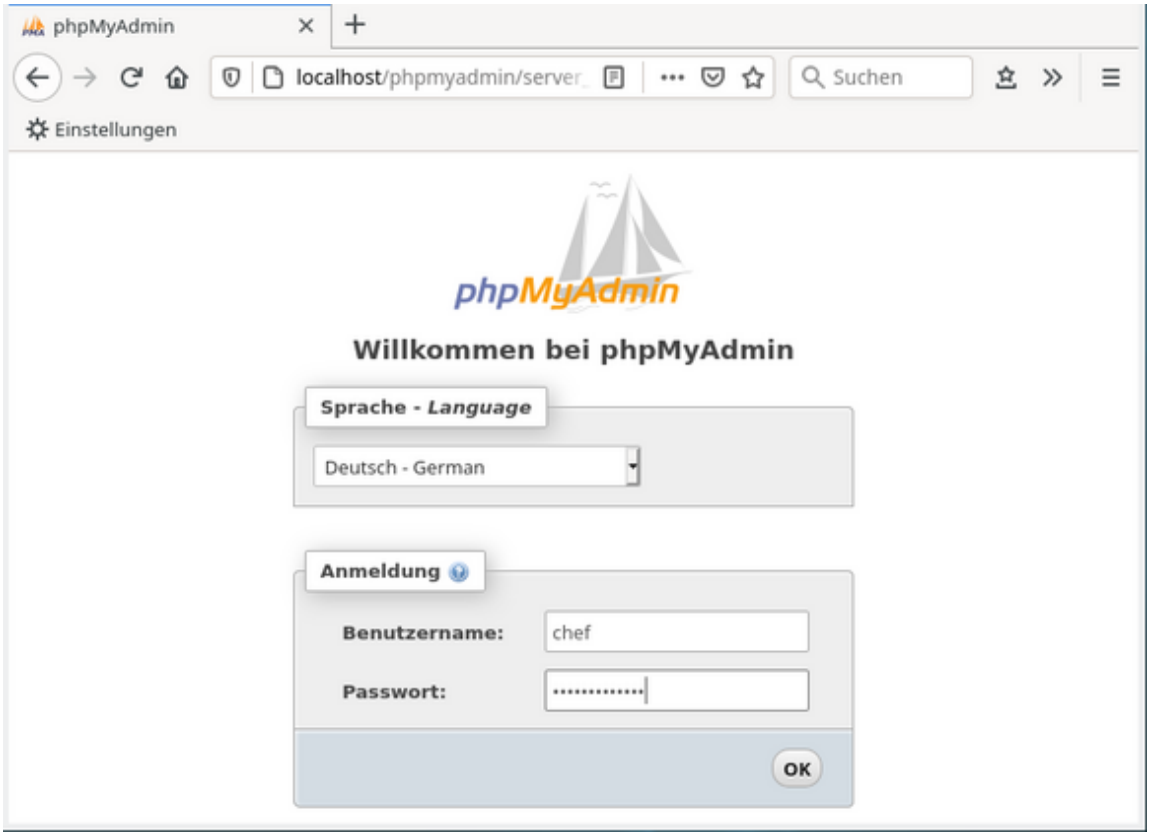


Abbildung 43: Loginfenster

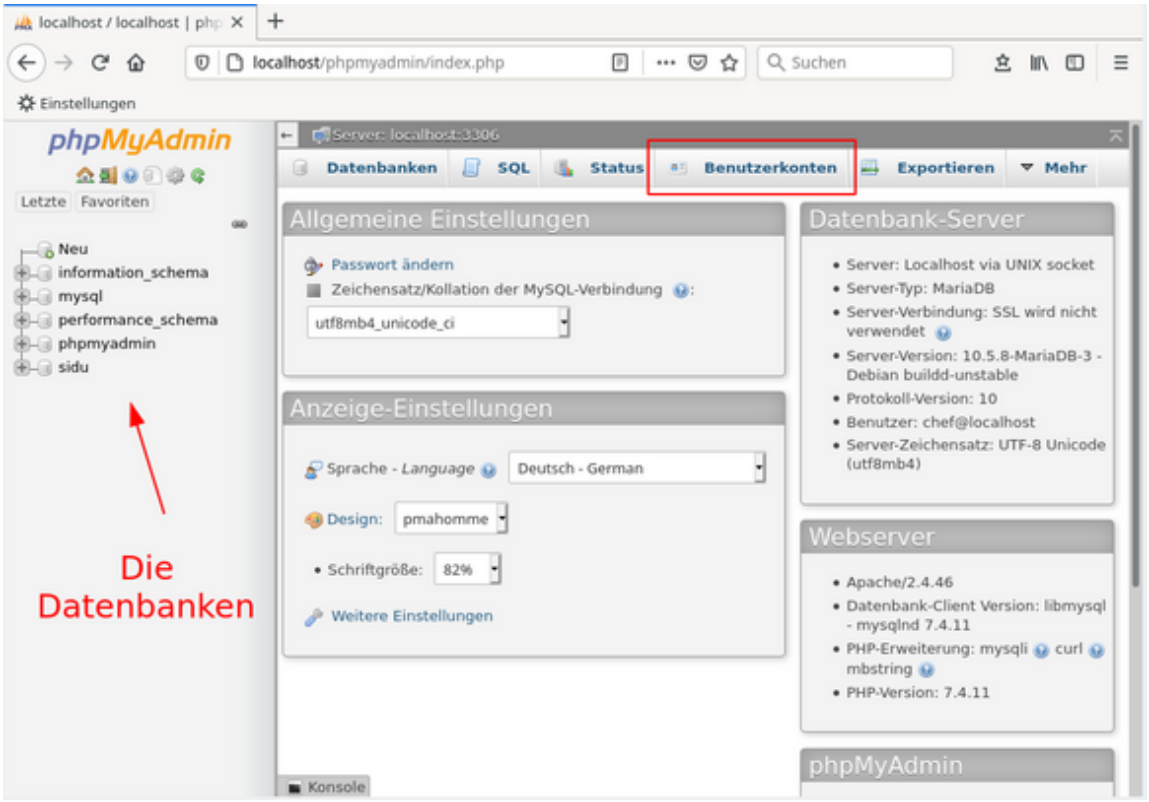


Abbildung 44: Startfenster

Abbildung 45: Benutzerkonten

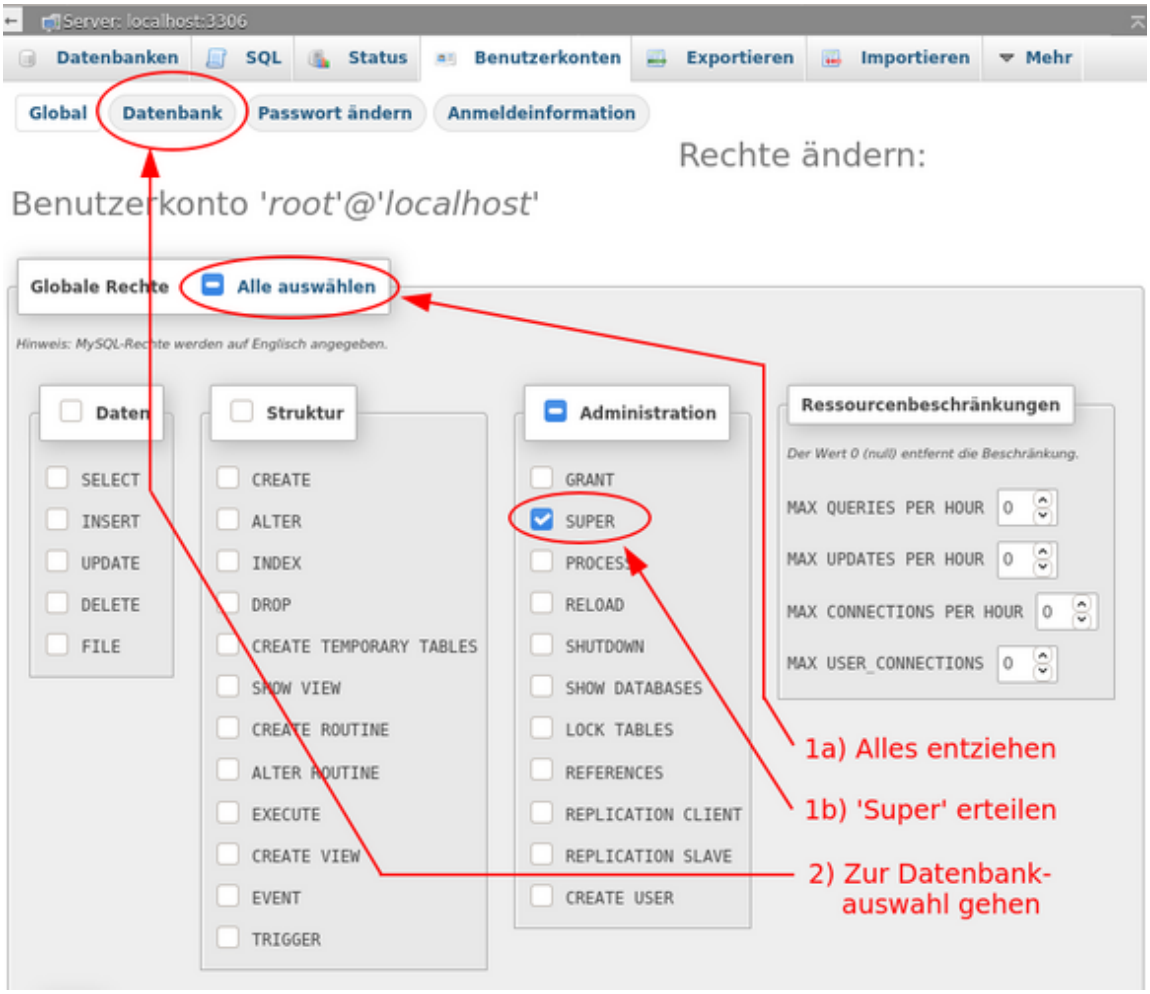


Abbildung 46: Rechte eines Benutzers verwalten (1a, 1b)

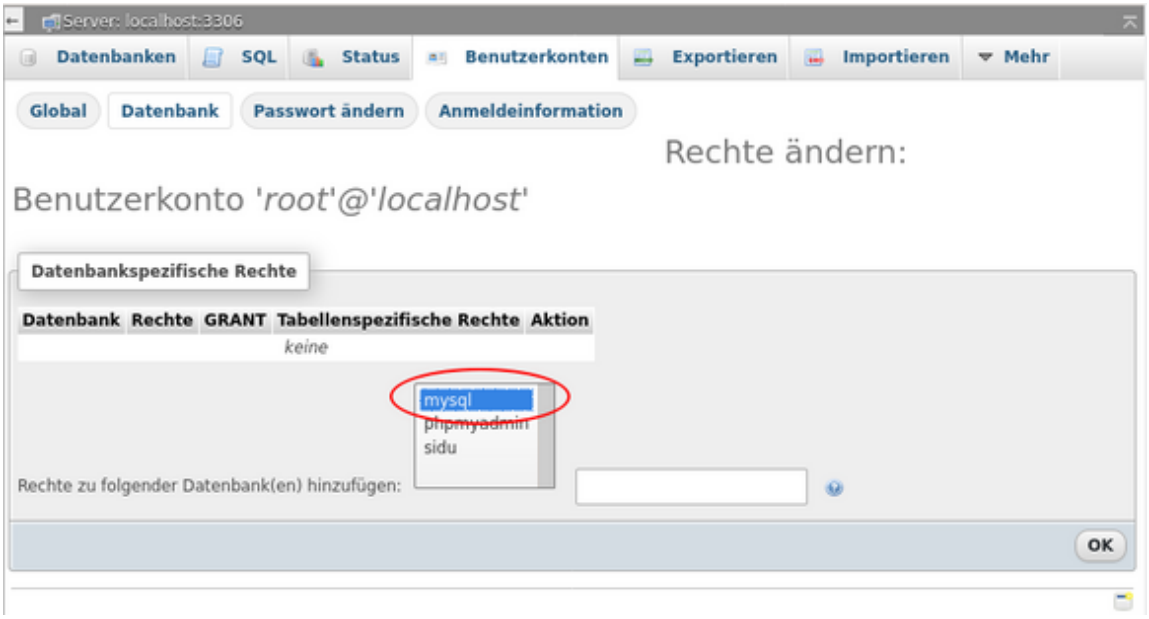


Abbildung 47: Rechte eines Benutzers verwalten (2)

Ausgewählt wird ausschließlich die Methode *"SELECT"*. Ein Klick auf `OK` führt den sql-Befehl aus.

Somit sind wir an Ziel und verlassen *phpMyAdmin* über das in der linken Spalte platzierte Tür-Icon.

phpMyAdmin bietet umfangreiche Möglichkeiten zur Verwaltung der Datenbanken, ihrer Tabellen und deren Inhalte. Beachtet werden sollte der Reiter `Exportieren` im Hauptfenster, hinter dem sich die Möglichkeit zur Datensicherung findet.

5.8.5 Integration in Systemd

Die Steuerung von MariaDB wurde in Debian, und damit auch in siduction, in den Systemd integriert. MariaDB startet automatisch beim Booten des Servers. Die Steuerungsaufrufe lauten:

```
# systemctl [start | stop | restart] mariadb.service
```

Start- und Fehlermeldungen des Servers fließen in das Systemd Journal ein. Genaue Informationen enthält die externe Webseite MariaDB Systemd.

Bei Suchanfragen im Internet zur Systemsteuerung von MariaDB sollte darauf geachtet werden, dass sich die Fundstellen auf Systemd beziehen.

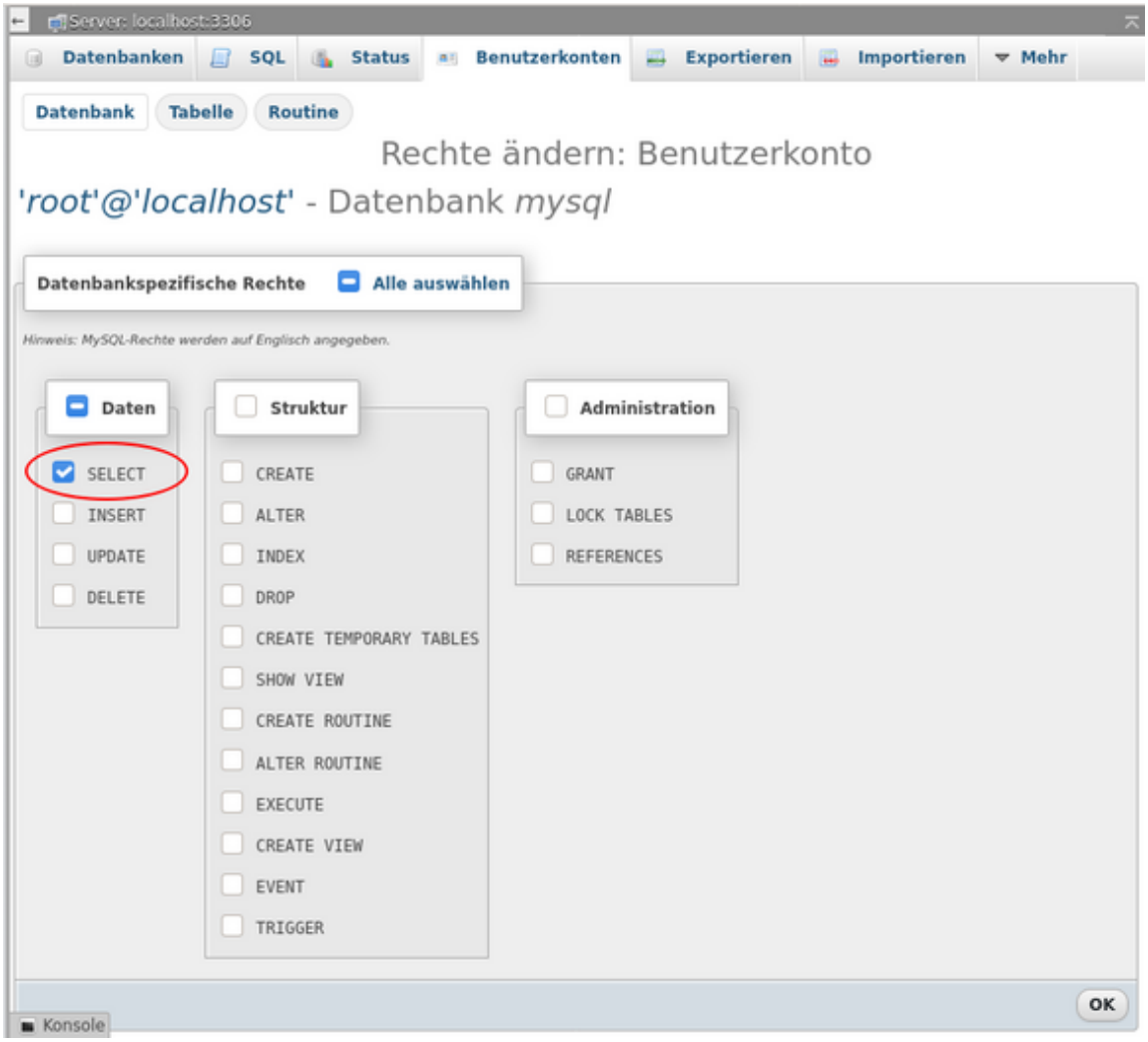


Abbildung 48: Rechte eines Benutzers verwalten (DB mysql)

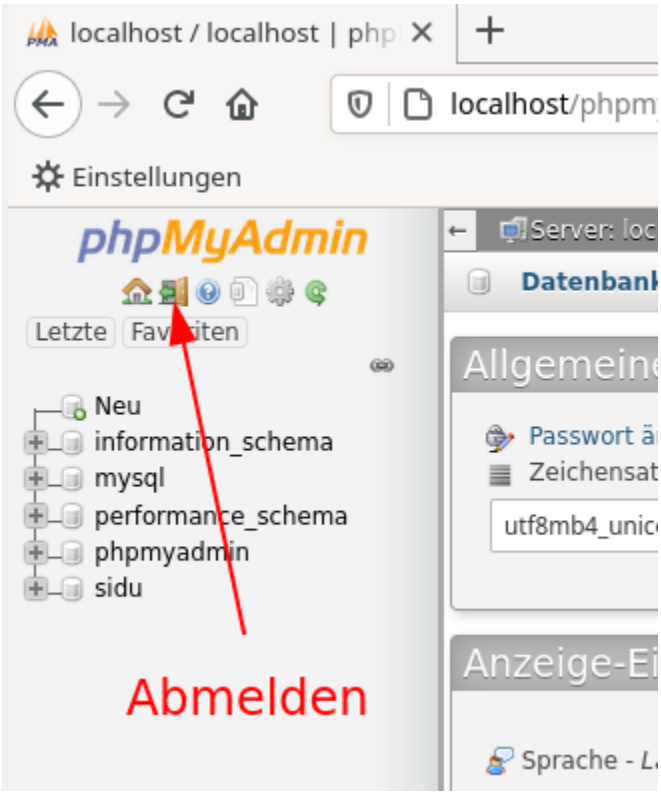


Abbildung 49: phpMyAdmin beenden

5.8.6 MariaDB Log

Das systemd Journal enthält Meldungen über den Startprozess der Unit `mariadb.service`. Es ist die erste Anlaufstelle wenn Fehler auftreten.
In der Konsole zeigt der Befehl `journalctl` die Meldungen zu MariaDB mit:

```
journalctl -n 25 -u mariadb.service
```

z.B. die letzten 25 Zeilen.

Oder fortlaufend mit:

```
journalctl -f -u mariadb.service
```

Darüber hinaus schaltet man das Loggen der sql-Aktionen im MariDB-CLI so ein:

```
MariaDB [(none)]> SET GLOBAL general_log=1;
```

Das erstellt eine Log-Datei nach dem Muster `<Host>.log` im Verzeichnis `/var/lib/mysql/`.

Achtung: Dies ist ein absoluter Performence-Killer und nur dazu gedacht um kurzfristig die Aktionen zu beobachten.

5.8.7 Quellen MariaDB

MariaDB Dokumentation (englisch)
MariaDB Systemd (englisch)
und die Manpage

Die mariadb Manpage `man mariadb`

phpMyAdmin Dokumentation (deutsch)

Zuletzt bearbeitet: 2020-11-29

5.9 PHP einrichten

PHP ist in siduction nach der Installation mit der standardmäßigen Konfiguration sofort einsatzfähig.

5.9.1 PHP im Dateisystem

Debian hat die Dateien von PHP entsprechend ihrer Funktion vollständig in das Dateisystem integriert.

- In `/usr/bin/` das ausführbare Programm `php7.x`
  - und der Link `php`, der über `/etc/alternatives/php` auf `/usr/bin/php7.x` verweist.

- In `/usr/lib/php/` die installierten Module.

- In `/usr/share/php/` und `/usr/share/php\<Modul\>` gemeinsam genutzte Programmteile und Module.

- In `/etc/php/` die Konfigurationsverzeichnisse und -dateien.

- In `/var/lib/php/` der zur Laufzeit aktuelle Zustand der Module und Sessions.

5.9.2 PHP-Unterstützung für Apache2

Standardmäßig lädt der Apache Webserver die Unterstützung für PHP. Wir überprüfen das mit:
(dabei ist im Folgenden das x mit der Ziffer entsprechend dem Minor-Attribut der aktuell verwendeten PHP-Version zu ersetzen, also etwa 7.4)

```
# ls /etc/apache2/mods-enabled/* | grep php
/etc/apache2/mods-enabled/php7.x.conf
/etc/apache2/mods-enabled/php7.x.load
```

und erkennen, dass Apache das PHP-Modul für die Version 7.x geladen hat.
Damit der PHP-Interpreter veranlasst wird, Dateien mit der Endung *".php"* zu

verarbeiten, muss in der Apache Konfigurationsdatei `dir.conf` die Direktive `DirectoryIndex` den Wert `index.php` enthalten. Auch das prüfen wir:

```
# cat /etc/apache2/mods-available/dir.conf
<IfModule mod_dir.c>
    DirectoryIndex index.html index.cgi index.pl index.php ⇙
       index.xhtml index.htm
</IfModule>
```

Der Verwendung von PHP steht nichts im Wege, denn wir sehen das der Wert *“index.php”* enthalten ist.

5.9.3 PHP Konfiguration

Das Verzeichnis `/etc/php/7.x/` enthält die Konfiguration geordnet nach den zur Verfügung stehenden Interfaces.
Die Ausgabe zeigt den Zustand nach der Erstinstallation.

```
# ls -l /etc/php/7.x/
insgesamt 20
drwxr-xr-x 3 root root 4096 18. Dez 16:54 apache2
drwxr-xr-x 3 root root 4096 18. Dez 16:54 cli
drwxr-xr-x 2 root root 4096 18. Dez 16:54 mods-available
```

Mit den weiter unten installierten Modulen *“php7.x-cgi”* und *“php7.x-fpm”* sind zwei neue Verzeichnise hinzugekommen.

```
# ls -l /etc/php/7.x/
insgesamt 20
drwxr-xr-x 3 root root 4096 18. Dez 16:54 apache2
drwxr-xr-x 3 root root 4096  1. Feb 21:23 cgi
drwxr-xr-x 3 root root 4096 18. Dez 16:54 cli
drwxr-xr-x 4 root root 4096  1. Feb 21:23 fpm
drwxr-xr-x 2 root root 4096  1. Feb 13:22 mods-available
```

Jedes der Verzeichnisse *“apache2”*, *“cgi”*, *“cli”* und *“fpm”* enthält einen Ordner `conf.d` und eine Datei `php.ini`.
Die jeweilige *“php.ini”* beinhaltet die Konfiguration für das entsprechende Interface

und kann bei Bedarf geändert oder ergänzt werden. Der Ordner *“conf.d”* enthält die Links zu den aktivierten Modulen.

5.9.4 PHP Module

Abfragen

Für PHP steht eine Vielzahl von Modulen zu Verfügung. Welche bereits installiert wurden, erfährt man mit:

```
# dpkg-query -f='${Status}\ ${Package}\n' -W php7.4* | grep ↙
   '^install'
install ok installed php7.4
install ok installed php7.4-bz2
install ok installed php7.4-cli
install ok installed php7.4-common
install ok installed php7.4-curl
install ok installed php7.4-gd
install ok installed php7.4-imagick
install ok installed php7.4-json
install ok installed php7.4-mbstring
install ok installed php7.4-mysql
install ok installed php7.4-opcache
install ok installed php7.4-readline
install ok installed php7.4-xml
install ok installed php7.4-zip
```

Um verfügbare, aber nicht installierte Module anzuzeigen, schreiben wir die Suche am Ende etwas um:

```
# dpkg-query -f='${Status}\ ${Package}\n' -W php7.4* | grep '↙
   not-install'
unknown ok not-installed php7.4-calendar
unknown ok not-installed php7.4-cgi
unknown ok not-installed php7.4-ctype
unknown ok not-installed php7.4-dom
unknown ok not-installed php7.4-exif
unknown ok not-installed php7.4-ffi
unknown ok not-installed php7.4-fileinfo
unknown ok not-installed php7.4-fpm
```

```
unknown ok not-installed php7.4-ftp
unknown ok not-installed php7.4-gettext
unknown ok not-installed php7.4-iconv
unknown ok not-installed php7.4-pdo
unknown ok not-installed php7.4-pdo-mysql
unknown ok not-installed php7.4-phar
unknown ok not-installed php7.4-posix
unknown ok not-installed php7.4-shmop
unknown ok not-installed php7.4-simplexml
unknown ok not-installed php7.4-sockets
unknown ok not-installed php7.4-sysvmsg
unknown ok not-installed php7.4-sysvsem
unknown ok not-installed php7.4-sysvshm
unknown ok not-installed php7.4-tokenizer
unknown ok not-installed php7.4-xsl
```

Jetzt kennen wir die genauen Bezeichnungen der Module.

Info

Ausführlichere Beschreibungen zu den Modulen liefert der Befehl

```
# apt show <Modulname>
```

Installation

Um Module zu installieren verwenden wir z.B.:

```
# apt install php7.x-cgi php7.x-fpm
```

Die beiden Module unterstützen CGI-Scripte und Fast/CGI Requests.
Anschließend starten wir den Apache neu:

```
# systemctl restart apache2.service
```

Handling

Der Zustand der PHP-Module ist während der Laufzeit veränderbar. Das ermöglicht auch die Steuerung von Modulen in Scripten um sie vor der Verwendung zu laden und nachher wieder zu entladen.

- `phpenmod` – aktiviert Module in PHP
- `phpdismod` – deaktiviert Module in PHP
- `phpquery` – Zeigt den Status der PHP Module

Nicht benötigte Module (im Beispiel imagick) deaktiviert in der Konsole der Befehl

```
# phpdismod imagick
```

Um das Modul imagick für alle Iterfaces zu laden, dient der Befehl

```
# phpenmod imagick
```

Verwenden wir die Option `-s apache2`

```
# phpenmod -s apache2 imagick
```

wird das Modul nur für Apache2 geladen.

Die Statusabfrage mit `phpquery` erfordert immer die Angabe der Modulversion und des Interface. Hier einige Beispiele:

```
# phpquery -v 7.4 -s apache2 -m zip
zip (Enabled for apache2 by maintainer script)

# phpquery -v 7.4 -s cli -m zip
zip (Enabled for cli by maintainer script)

# phpquery -v 7.4 -s fpm -m zip
zip (Enabled for fpm by maintainer script)

# phpquery -v 7.4 -s apache2 -m imagick
imagick (Enabled for apache2 by local administrator)
```

Bei dem Modul imagick zeigt uns der String *“Enabled for apache2 by local administrator”*, dass es nicht wie das zip-Modul automatisch beim Start geladen wurde, sondern dass der Administrator es manuell aktiviert hat. Die Ursache liegt in den zuvor benutzten Befehlen `phpdismod` und `phpenmod` für dieses Modul.

5.9.5 Apache Log

Der Apache Server speichert die Fehlermeldungen von PHP in seinen Log-Dateien unter `/var/log/apache2/`. Gleichzeitig erscheint bei fehlerhaften PHP-Funktionen eine Meldung in der aufgerufenen Webseite.
Alternativ lassen wir uns die Log-Funktionen anzeigen.

```
# php --info | grep log
[...]
error_log       => no value
log_errors      => On
log_errors_max_len  => 1024
mail.log      => no value
opcache.error_log   => no value
[...]
```

In den Dateien `/etc/php/7.x/<Interface>/php.ini` haben wir die Möglichkeit die nicht gesetzten Werte durch eigene, tatsächlich vorhandenen Logdateien zu ersetzen.

5.9.6 Quellen PHP

PHP - deutsches Handbuch
PHP - aktuelle Meldungen
tecadmin - Modulhandling (englisch)

Zuletzt bearbeitet: 2021-07-20

6 Hardware

6.1 Grafiktreiber

für nVidea, Intel, ATI/AMD

Wir gehen hier im Handbuch nur auf die verbreitetsten Grafikkarten ein. Exotische oder relative alte Grafikhardware, sowie Server-Grafik findet hier keine Beachtung.

6.1.1 Open Source Xorg-Treiber

Welche Grafikhardware verbaut ist erfährt man relativ einfach

```
inxi -G
lspci | egrep -i "vga|3d|display"
```

Diese Information ist auch überaus wichtig, sollte man Probleme mit der Grafik haben und Hilfe im Forum oder dem IRC suchen.

Das Grafiksystem unter Linux besteht aus 4 grundlegenden Teilen:

- Kernel Treiber
  - radeon/amdgpu (ATI/AMD Grafik)
  - i915 (Intel Grafik)
  - nouveau (nVidia Grafik)
- Direct Rendering Manager
  - libdrm-foo
- DDX Treiber
  - xserver-xorg-video-radeon/amdgpu
  - xserver-xorg-video-intel
  - xserver-xorg-video-nouveau

  Xorg kann auch den modesetting-ddx verwenden, welcher mittlerweile Bestandteil des Xservers selbst ist. Dieser wird automatisch für Intel Grafik benutzt und auch dann, wenn kein spezielles xserver-xorgvideo-foo Paket installiert ist.
- dri/mesa

- libgl1-mesa-glx
- libgl1-mesa-dri
- libgl1-mesa-drivers *Dieser Teil von Xorg ist die freie OpenGL Schnittstelle für Xorg.*

Open Source Xorg-Treiber für nVidia (modesetting/nouveau), ATI/AMD (modesetting/radeon/amdgpu), Intel (modesetting/intel) und weitere sind mit siduction vorinstalliert.

Anmerkung: xorg.conf wird für Open-Source-Treiber in der Regel nicht mehr benötigt Ausnahmen sind z.B. Mehrschirmbetrieb.

6.1.2 Proprietäre Treiber

Proprietäre Treiber gibt es faktisch nur noch für nVidia Grafikkarten. AMD hat zwar auch einen proprietären Treiber namens amdgpu-pro, dieser unterstützt aber offiziell nur Ubuntu in bestimmten Versionen und liegt in Debian nicht paketiert vor. Außerdem ist dieser Treiber eher für professionelle Karten denn für Desktop Karten konzipiert.

Um vom proprietären Treiber von Nvidia auf nouveau zu wechseln, siehe den Eintrag im siduction Wiki.

Hier erhältst du mehr Informationen zu den Treibern von
Intel
ATI/AMD
nouveau
X.Org.

6.1.3 Videotreiber 2D

So ziemlich jede Grafikkarte, welche einen KMS Treiber kernelseitig benutzt, ist für den 2D Betrieb unter allen Oberflächen geeignet. In aller Regel (bis auf wenige Ausnahmen exotischer oder alter Hardware) ist auch 3D Beschleunigung vorhanden.

6.1.4 Videotreiber 3D

3D Beschleunigung steht unter Linux für Intel-, AMD- und nVidia-Grafikkarten zur Verfügung. Wie gut die freien Treiber 3D implementiert haben, hängt ein wenig von der Grafikkarte selbst ab. Generell ist anzumerken, dass fast alle Grafikkarten nicht-freie Firmware benötigen, um einen problemlosen Betrieb zu ermöglichen. Diese Firmware gibt es bei Debian nur im non-free Repository, da sie nicht DFSG konform ist. Ist die korrekte Firmware installiert, ist 3D Support mit Intel oder AMD Grafikkarten ohne weiteres Zutun verfügbar. Bei nVidia Grafik sieht die Geschichte etwas anders aus. Ältere Karten, welche seitens nVidia als legacy Karten eingestuft sind, funktionieren relativ gut, auch wenn immer mit Problemen zu rechnen ist, da auch der verwendete Desktop eine Rolle spielt. Der freie nouveau-Treiber wird ohne Unterstützung von nVidia per reverse engineering entwickelt.

Da für den korrekten Betrieb in der Regel (AMD, Intel ab Skylake und Nvidia ab Fermi) die nicht-freie Firmware benötigt wird, sollte in /etc/apt/sources.list/debian.list ein Eintrag analog

```
deb http://deb.debian.org/debian/ unstable main contrib non-free
```

gesetzt sein. Um sich nachfolgende Probleme mit WLAN, Netzwerk, Bluetooth oder Ähnliches zu ersparen, ist ein

```
apt update && apt install firmware-linux-nonfree
```

sinnvoll. Damit installiert man zwar mehr Firmwares, als man evtl. benötigt, das sollte aber kein Nachteil sein.

6.1.5 nVidia closed Source Treiber

Auswahl, Installation mit dkms-Unterstützung und Integration in xorg

nVidia teilt seine Grafikkarten-Treiber in 7 Generationen auf:

1. Riva TNT, TNT2, GeForce, und einige GeForce 2000 GPUs
2. GeForce 2000 bis GeForce 4000 series GPUs
3. GeForce 5000 series GPUs
4. GeForce 6000 and 7000 series GPUs
5. GeForce 8000 and 9000 series GPUs

6. GeForce 400 und 500 series GPUs (Fermi GF1xx)
7. Geforce 600, 700, 800 (Kepler GK1xx GK2xx, Maxwell GM1xx GM2xx,); Geforce 10xx (Pascal GP1xx), Geforce 16xx/20xx (Turing TU1xx); Geforce 30xx (Ampere GA1xx)

Karten der Generationen 1 - 5 werden seitens nVidia nicht mehr unterstützt, es gibt hierfür nur alte Treiber-Versionen, die weder mit aktuellen Kerneln, noch mit aktuellen Versionen des Xorg-Servers funktionieren. Für eine komplette und aktuelle Liste unterstützter Grafikchips konsultiere bitte “Supported Products List” auf der Downloadseite für NVIDIA-Linux Grafiktreiber.

Debian stellt folgende Versionen der binären Treiber zur Verfügung:

```
- nvidia-legacy-304xx-driver (für 4.)
- nvidia-legacy-340xx-driver (für 5.)
- nvidia-legacy-390xx-driver (für 6.)
- nvidia-driver (für 7.)
```

Da es sich hier aber um proprietäre Treiber handelt, muss in den Sources contrib und non-free aktiviert sein (wie auch für die Firmware für freie Treiber). Es ist im Vorfeld sicher zu stellen, dass die kernel-header passend zum laufenden Kernel installiert sind. Das ist der Fall, sobald linux-image-siduction-amd64 und linux-headers-siduction-amd64 installiert sind. Außerdem sind die Pakete gcc, make und dkms notwendig. Mit dkms werden zusätzlich installierte (nVidia-)Kernelmodule automatisch bei einem Kernelupdate aktualisiert. Nachdem man nun mit den genannten Befehlen herausgefunden hat, welche nVidia Karte, bzw welchen nVidia Chip man hat, kann man den Treiber wie folgt installieren:

GeForce 8000 and 9000 series

```
apt update && apt install nvidia-legacy-340xx-driver
```

GeForce GF1xx Chipsatz, Fermi Cards

```
apt update && apt install nvidia-legacy-390xx-driver
```

Kepler, Maxwell, Pascal und neuer (GKxxx, GMxxx, GPxxx, TU1xx)

```
apt update && apt install nvidia-driver
```

Wenn das fehlerfrei durchgelaufen ist noch ein

```
mkdir -p /etc/X11/xorg.conf.d; echo -e 'Section "Device"\n\↙
   tIdentifier "My GPU"\n\tDriver "nvidia"\nEndSection' > /↙
   etc/X11/xorg.conf.d/20-nvidia.conf
```

ausführen, um Xorg mitzuteilen, diesen installierten Treiber zu benutzen. Nach einem Reboot sollte das System hoffentlich bis in den Desktop starten. Sollten Probleme auftreten, sprich der Desktop nicht starten, so sollte man /var/log/Xorg.0.log konsultieren.

Da die Legacy Treiber 304.xx und 340.xx von NVidia nicht mehr supportet werden, ist damit zu rechnen, dass selbige mit einem neuen Kernel oder neuem Xorg nicht mehr funktionieren.

Problematisch sind Notebooks mit Hybridgrafik Intel/nVidia, sogenannte Optimus Hardware. Hier wurde früher auf Bumblebee verwiesen, diese Lösung ist aber alles Andere, als optimal. nVidia selbst empfielt hingegen diese Setups per PRIME zu konfigurieren. Unsere Empfehlung ist aber, solche Hardware, wenn es geht, zu vermeiden. Tipps zur Einrichtung für Optimus Hardware können wir hier nicht geben.

Zuletzt bearbeitet: 2021-05-05

7 Systemadministration

Dieser Abschnitt beinhaltet Informationen und Hinweise zur/zum

- Terminal - Kommandozeile, grundlegende Einführung, Arbeit als root, farbiges Terminal, Hilfe im Terminal und Skripte benutzen.
- Doas - Alternative zu Sudo, unsere Empfehlung für all jene, die sudo vermissen.
- Systemadministration allgemein Kurz und knapp, ein Streifzug durch die Systemadministration; Bootoptionen, systemd - Dienste verwalten, Prozesse beenden, Passwörter verwalten, Schriftarten in siduction, das Drucksystem CUPS und Sound in siduction.
- APT Paketverwaltung, Paketquellen, Pakete verwalten, Aktualisierung des Systems, Programmpakete suchen und warum ausschließlich apt verwendet werden soll.
- Lokaler APT-Mirror; Apt-Cacher, der Proxy-Server für Debian-Pakete. Server installieren und die Client Konfiguration.
- Neue Kernel installieren, Kernel-Aktualisierung ohne Systemaktualisierung, 3rd Party Module und alte Kernel entfernen.
- Systemd der System- und Dienste-Manager, Konzeption des systemd, Unit Typen, Systemd im Dateisystem und die Handhabung von Diensten.
  - Die systemd unit-Datei, Verzeichnisse und Hierarchien der Unit-Dateien, die Eingliederung in systemd, der Aufbau der Unit-Datei mit Beschreibung zahlreicher Optionen, die Funktion der Unit-Dateien am Beispiel von CUPS und die Werkzeuge, die systemd bereitstellt,
  - systemd-service Unit, eine service-Unit anlegen und die Beschreibung aller wesentlichen Optionen.
  - systemd-mount Unit, Inhalt der mount-Unit, Inhalt der automount-Unit, Namenskonventionen, Einsatzbereiche und einige Beispiele.
  - systemd-target - Ziel-Unit, von Runlevel zu systemd-target, zu berücksichtigende Besonderheiten.

  - systemd-path Unit, die benötigten Dateien, die Optionen der path-Unit, path-Unit anlegen und eingliedern und das Beispiel “Überwachung von DocumentRoot des Apache Webservers”.
  - systemd-timer Unit, die benötigten Dateien, die Optionen der timer-Unit, timer-Unit anlegen und eingliedern, timer-Unit als cron Ersatz.

- Systemjournal, der journald lokal und über das Netzwerk, journald konfigurieren, Abfrage des systemd-Journals mit journalctl, die Ausgaben filtern und steuern, Beispiele um journalctl zu beherrschen.

Zuletzt bearbeitet: 2022-03-01

7.1 Terminal - Kommandozeile

Ein Terminal, auch Konsole genannt, ist ein Programm, das es einem ermöglicht, durch direkt ausgeführte Befehle unmittelbar mit dem GNU/Linux Betriebssystem zu interagieren. Das Terminal, auch häufig die *“Shell”* oder *“Kommandozeile”* genannt, ist ein äußerst mächtiges Werkzeug und den Aufwand wert, die Grundlagen seiner Handhabung zu erlernen.

In siduction kann man das Terminal/die Konsole aufrufen, indem man das PC-Monitorsymbol rechts des Menüs anklickt oder im *Menü* > *System* > *Terminal* aufruft, oder, noch einfacher, in die Suchleiste des Menü *“kons”* oder *“term”* eintippt.

Nach dem Aufrufen des Terminals sieht man die Eingabeaufforderung “prompt”:

```
username@hostname:~$
```

“username” in obigem Beispiel entspricht dem Nutzernamen des angemeldeten Benutzers. Die Tilde `~` zeigt, man befindet sich in seinem Heimverzeichnis `/home/username` , und das Dollarzeichen (der Promt) `$` bedeutet, dass man im Terminal mit eingeschränkten Benutzerrechten angemeldet ist. Am Ende blinkt der Cursor. Dies alles ist die Kommandozeile. Hier werden Befehle eingegeben, die das Terminal ausführen soll.

Viele Befehle kann man nur mit root-Rechten, also Administratorrechten, ausführen. Root-Rechte erhält man, indem man `su` eingibt und `Enter` drückt. Hiernach muss man das Rootpasswort eingeben. Das Passwort wird während der Eingabe auf dem Bildschirm nicht angezeigt. (Siehe unten Arbeit als root)

Ist die Eingabe korrekt, zeigt die Kommandozeile nun:

```
root@hostname:/home/username#
```

Zu beachten ist, dass das Dollarzeichen `$` durch eine Raute `#` ersetzt wurde. In einem Terminal bedeutet die Raute immer, dass man mit **root**-Rechten angemeldet ist.
Wenn im Handbuch Kommandozeilenbefehle angegeben werden, werden die Angaben vor dem Prompt ($ oder #) ausgelassen. Ein Befehl wie:

```
# chmod g+w <Datei>
```

bedeutet also: man öffnet ein Terminal, meldet sich als **root** an (`su`) und führt dann den Befehl an einem Rootprompt `#` aus. Die Raute wird nicht mit eingegeben.

Ein weiterer Hinweis:
Für User, die neu am Terminal arbeiten, ist es oft verwirrend, wenn nach dem Ausführen eines Befehls keine Meldung erscheint, sondern nur wieder der leere Prompt. Diese Funktion ist gewollt und bedeutet, dass der Befehl fehlerfrei ausgeführt wurde. (Im obigen Beispiel erhielten die Gruppenmitglieder Schreibrechte an der <Datei>.)

7.1.1 Arbeit als root

Achtung:
Während man mit root-Rechten im Terminal eingeloggt ist, darf man alles, z. B. Dateien löschen, ohne die das Betriebssystem nicht mehr funktioniert, uvm. Wenn man mit root-Rechten arbeitet, muss man sich darüber im Klaren sein, *was* man gerade macht, denn es ist leicht möglich, dem Betriebssystem irreparable Schäden zuzufügen.

Berücksichtigen muss man, dass alle Aktionen, soweit im Programm vorgesehen, auch mit root-Rechten ausgeführt werden. Der einfache copy-Befehl **`cp <Quelle> <Ziel>`** in einem User-Verzeichnis führt zu Dateien mit dem Eigentümer **root** im Zielverzeichnis. Das ist vermutlich nicht gewollt und auch nicht sinnvoll.

Deshalb: **Arbeiten als Root nur dort wo es wirklich notwendig ist!**

Über su

Eine Anzahl von Befehlen muss mit root-Rechten gestartet werden. Diese Rechte erhält man durch Eingabe von **`su`**. Nach der Eingabe des richtigen Passwortes erscheint der Root-Prompt.

```
$ su
Passwort:
#
```

Jetzt ist es möglich im Terminal alle Befehle auszuführen und alle Programme zu starten, die root-Rechte erfordern. Beenden kann man diesen Status mit der Eingabe von

```
# exit
$
```

und es erscheint wieder der Promt für den User.

Über su-to-root

Im Gegensatz zum allgemeinen Befehl *“su”* erlaubt *“su-to-root”* das Ausführen von Programmen mit graphischer Oberfläche mit root-Rechten. *“su-to-root”* transferiert unter Benutzung von *“su”* die X-Eigenschaften an den Zielnutzer. Die Eingabe lautet:

```
su-to-root -X -c <Programm>
```

Wenn Fehlermeldungen mit Bezug zu *“dbus”* auftreten, ist die Eingabe zu erweitern:

```
su-to-root -X -c 'dbus-launch <Programm>'
```

Es öffnet sich ein weiteres Terminal, in das das root-Passwort einzugeben ist. Bei Erfolg startet das gewünschte Programm mit root-Rechten.
Beispiele für die Verwendung graphischer Anwendungen mittels *“su-to-root”* sind: Die Bearbeitung einer Konfigurationsdatei mit einem Texteditor, der Einsatz des Partitionierungsmanagers gparted oder die Verwendung von Dateimanagern wie dolphin oder thunar.

Verwendung in den Desktopumgebungen:

- Plasma (KDE und LXQt)
  Der Befehl ist in Plasma nicht notwendig und wird nicht unterstützt, denn für Programme, die root-Rechte benötigen erfolgt eine Passwortabfrage und beim Editor erfolgt die Abfrage wenn man die geänderte Datei speichern möchte. Deshalb nur `su` im Terminal verwenden, wenn nötig.
- Gnome und Cinnamon
  Das Verhalten ist dem in Plasma ähnlich, mit der Ausnahme, dass der Befehl (su-to-root) unterstützt wird, aber nicht notwendig ist.
- XFCE und Xorg
  Hier entfaltet der Befehl seine volle Macht, und man ist in der Lage das gewünschte graphische Programm mit root-Rechten zu starten. Mann ist je-

doch auch in der Pflicht zu beachten, wann und mit welchem Programm root-Rechte wirklich erforderlich sind.

Unter keinen Umständen sollten Produktivprogramme, die normalerweise mit Benutzerrechten gestartet werden, mit dieser Option als root hochgefahren werden: Internet-Browser, E-Mail-Programme, Büroprogramme u.a.

sudo ist nicht konfiguriert

Sudo steht nur im Live-Modus zur Verfügung, da im Live-Modus kein Root-Passwort gesetzt ist.
Nach einer Installation ist sudo nicht aktiviert. Der Grund ist: Sollte ein Angreifer das Nutzer-Passwort abgreifen, erlangt er noch keine Super-User-Rechte und kann keine schädlichen Veränderungen am System durchführen.
Ein anderes Problem mit sudo ist, dass eine Root-Anwendung, die mit der Nutzerkonfiguration läuft, Berechtigungen ändern und somit für den Nutzer unbrauchbar machen kann. Die Verwendung von `su` oder `su-to-root` wird empfohlen!
Sollte man trotz aller Warnungen sudo nutzen wollen, so muss man den entsprechenden $user der Gruppe sudo hinzufügen!
Dies kann mit dem Befehl **`adduser BENUTZER GRUPPE`** als **root** ausgeführt werden.

Als schlanke Alternative zu *“sudo”* empfehlen wir `doas`. Die Handbuchseite Doas - Alternative zu Sudo erläutert die Vorteile von Doas und die Doas Konfiguration.

7.1.2 Farbiges Terminal

Farbige Prompt am Terminal können einen vor unangenehmen oder katastrophalen Fehlern bewahren, falls man als **root** eine Aufgabe durchführt, die man als **user** machen wollte.
Deshalb ist in siduction in der Grundeinstellung der Prompt des user grün, blau und weiß, und bei dem von root wird das Wort “root” in roter Farbe dargestellt.

Der Fokus beim Arbeiten mit dem Terminal sollte auf den Eingaben und Ausgaben der Befehle liegen und nicht auf bunte Prompt. In siduction haben wir uns trotzdem für die Farben entschieden, um den Usern einen Warnhinweis zu geben, wenn sie als Systemadministrator mit root-Rechten unterwegs sind.

Farbe des Prompt ändern

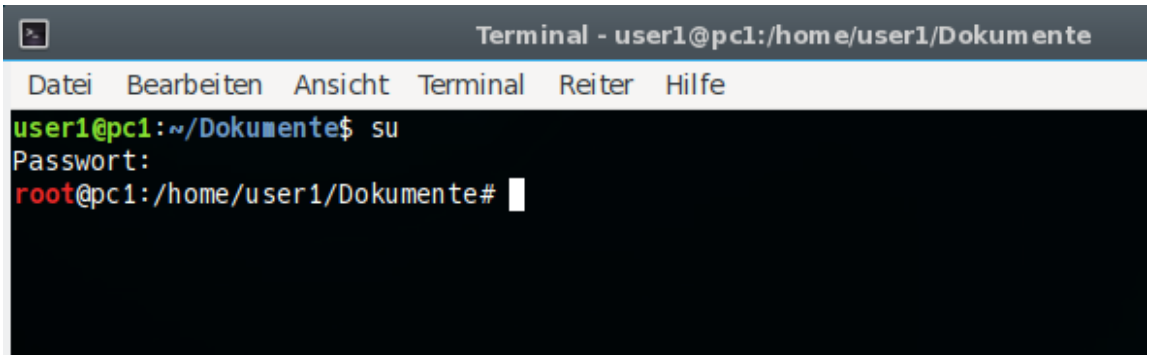


Abbildung 50: Farbiger Prompt

Bevor die Konfigurationsdatei geändert wird, erstellen wir im Terminal erst eine Sicherungskopie mit einem Datumsstempel.

```
$ cp ~/.bashrc ~/.bashrc_$(date +%F)
```

Dann öffnen wir mit einem Texteditor unserer Wahl (z. B.: kate, gedit, mcedit, vim…) die Datei `~/.bashrc` und suchen nach folgender Zeile, die sich etwa in der Mitte der Datei befindet:

```
PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h⤢
   \[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '
```

Dem Schrift- und Farbcode folgen unmittelbar die Promtteile, die diese Darstellung erhalten sollen. In der folgenden Abbildung wird der Bezug zwischen den Promtteilen und ihren Kürzeln gezeigt.

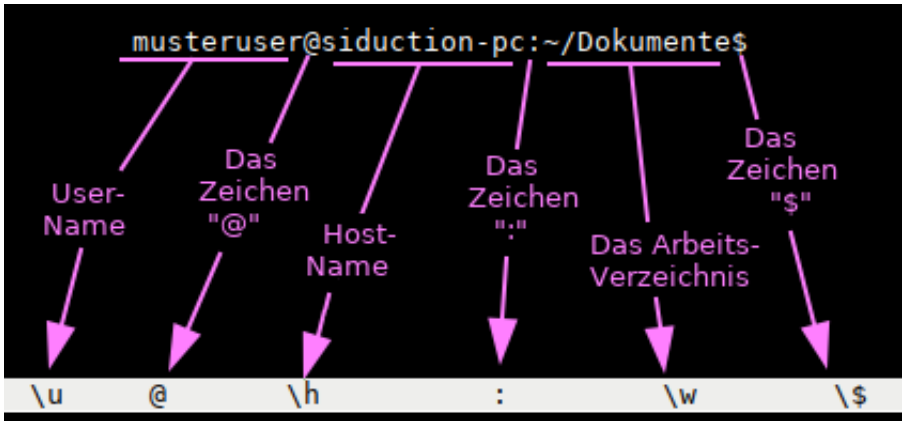


Abbildung 51: Prompt Code

Die anschließende Tabelle erklärt die Werte der Syntax “[\033[**01;32**m]”, wobei der fett gedruckte Teil die Schriftattribute und die Farbe bestimmt.

Schriftcode	Schriftattribut	Farbcode	Farbe
00m	Standard für Schrift und Farbe		
00;XX	Standardschrift	XX;m	Standardfarbe
01;XX	fett	XX;30	schwarz
02;XX	dunkel	XX;31	rot
03;XX	kursiv	XX;32	grün
04;XX	unterstrichen	XX;33	gelb
05;XX / 06;XX	blinkend	XX;34	blau
07;XX	block, invertiert	XX;35	magenta
08;XX	Hintergrundfarbe (unsichtbar)	XX;36	cyan
09;XX	durchgestrichen	XX;37	weiß

Die oben zitierte “PS1”-Zeile wird demnach wie folgt angezeigt:

Schriftcode	Promtteile und ihre Darstellung
[01;32m]\u@\h	user, @ und host erhalten die Attribute “fett” und “grün”
[00m]:	Doppelpunkt erhält die Standardattribute des Teminals
[01;34m]\w	das Arbeitsverzeichnis erhält die Attribute “fett” und “blau”
[00m]\$	der Prompt erhält die Standardattribute des Teminals

Soll aus dem Prompt die Farbe entfernt werden, stellen wir der PS1-Zeile eine Raute `#` und ein `Leerzeichen` voran. Damit ist die Zeile auskommentiert. Nun genügt es die Zeile

```
PS1='${debian_chroot:+($debian_chroot)}\[\033[00m\]\u@\h:\w\$↙
    '
```

unmittelbar als nächste Zeile einzufügen.

Soll in dem Prompt die Farbe geändert werden, ist für jeden Teil des Prompt die Farbkodierung anzupassen.

```
 PS1='${debian_chroot:+($debian_chroot)}\[\033[03;32m\]\u@\h↙
    \[\033[01;34m\]:\w\[\033[00m\]\$ '
```

Dieses Codebeispiel erzeugt einen Promt, in dem **Username** @ **Hostname** grün und kursiv; der **:** und das **Arbeitsverzeichnis** blau und fett; das **$**-Zeichen und die Befehlseingabe die Kontrastfarbe zum Hintergrund des Terminals erhalten.

Die neuen Farben und Formate erscheinen nach öffnen eines neuen Terminals.

Farbeinstellungen des Terminals

Im Menü des Terminals gibt es unter *Bearbeiten - Einstellungen…* - Reiter *Farben* eine Unmenge an Einstellungsmöglichkeiten. Wir empfehlen eine eher schlichte Einstellung.

7.1.3 Wenn das Terminal hängt

Manchmal kann ein Terminal nicht mehr so reagieren wie gewünscht. Das liegt meist daran, dass sich ein Programm fehlerhaft beendet und das Teminal in einem abnormalen Zustand zurückgelassen hat. Dann muss

```
reset
```

eingegeben und die Eingabetaste **Enter** gedrückt werden.

Wenn die Ausgabe eines Terminals verzerrt erscheint, kann dies meist durch das Drücken von **Strg**+**l** behoben werden, dadurch wird das Terminal-Fenster neu aufbaut. Solche Verzerrungen treten meist auf, wenn man mit Programmen arbeitet, eine eine ncurses-Schnittstelle benutzen, zum Beispiel *“cgdisk”*.

Ein Terminal kann eingefroren erscheinen, was aber in der Regel nicht der Fall ist, sondern die Eingaben werden weiterhin verarbeitet, auch wenn es nicht so scheint. Dies kann durch versehentliches Drücken von **Strg**+**s** verursacht sein. In diesem Fall kann **Strg**+**q** versucht werden, um die Konsole wieder frei zu geben.

7.1.4 Hilfe im Terminal

Die meisten Befehle/Programme haben eine Kommandozeilenhilfe und auch Anleitungen. Die Anleitungen werden im englischen *“man page”* oder *“manual page”*, im deutschen *“Manpage”* genannt. Die Syntax zum Aufrufen der man page ist:

```
$ man <Befehl>
```

oder

```
$ man -k <keyword>
```

Dies ruft die Manpage eines Befehls auf. Die Navigation in den Manpages erfolgt durch die Pfeiltasten, beendet werden sie mit `q` für quit. Beispiel:

```
$ man apt-get
```

Um eine Manpage zu verlassen, tippt man `q`

Ein anderes nützliches Werkzeug ist der `apropos` Befehl. Apropos ermöglicht es, die Manpages nach einem Befehl zu durchsuchen, wenn man z. B. die Syntax vergessen hat. Beispiel:

```
$ apropos apt-
```

Dies listet alle Befehle für den Paketmanager apt auf. Apropos ist ein mächtiges Werkzeug, für umfangreichere Informationen über apropos siehe

```
$ man apropos
```

7.1.5 Linux Konsolenbefehle

Eine sehr gute Einführung in die Konsole BASH findet sich auf linuxcommand.org(englisch).
Natürlich kann auch die favorisierte Suchmaschine verwendet werden, um mehr zu finden.

Brennen von CD, DVD und BD
Die Kommandozeilenprogramme bilden die Basis für beliebte GUI-Programme wie `K3b`, `Brasero` oder `Xfburn`. Wer die vollständige Bandbreite an Optionen der Kommandozeilenprogramme `cdrdao`, `wodim`, `growisofs` usw. bevorzugt, nutzt das Terminal. Unsere Handbuchseite DVD ohne GUI brennen enthält viele Beispiele und Tips um verfügbare Hardware zu erkennen, Daten zusammenzustellen und diese anschließend auf CD, DVD und BD zu brennen.

7.1.6 Skripte benutzen

Ein Konsolen-Skript ist ein bequemer Weg, um mehrere Befehle in einer Datei zu bündeln. Die Eingabe des Dateinamen des Skripts führt die Befehle, die im

Skript stehen, aus. siduction wird mit einigen sehr nützlichen Skripten ausgeliefert, welche Vereinfachungen der Systemadministration bieten.

Ein Skript wird in der Konsole folgendermaßen gestartet, wenn man sich im gleichen Verzeichnis befindet:

```
./name_des_skripts
```

Einige Skripte benötigen root-Zugang, abhängig vom Aufgabenbereich des Skripts.

Installation und Ausführung

Mit wget kann ein Skript auf den Rechner geladen werden, und man platziert es am besten in das empfohlene Verzeichnis, zum Beispiel nach `/usr/local/bin`. Zum Kopieren und Einfügen in der Konsole kann auch die Maus benutzt werden, nachdem man mit **su** root-Rechte erlangt hat.

Beispiel mit wget und root-Rechten

```
$ su
Passwort:
# cd /usr/local/bin
# wget -c ftp://<entfernter_server>/script-name.sh
```

Danach muss die Datei ausführbar gemacht werden:

```
# chmod +x script-name.sh
```

Da das Verzeichnis `/usr/local/bin` im Suchpfad von root enthalten ist, reicht für root der einfache Befehl

```
# script-name.sh
```

aus, um das Script zu starten.

Die Datei kann auch mit einem Browser auf den Computer geladen und an den geeigneten Ort verschoben werden, aber sie muss auch dann ausführbar gemacht werden.

Beispiel mit wget als Nutzer

So speichert man als Nutzer eine Datei im Verzeichnis `/home/<user>` (der Promt ist '$'):

```
$ wget -c ftp://<entfernter_server>/user-script-name.sh
$ chmod +x user-script-name.sh
```

Das Skript wird so gestartet:

```
$ ./user-script-name.sh
```

Das funktioniert als user natürlich nur, wenn das Script keine Befehle enthält, die root-Rechte benötigen.

Zuletzt bearbeitet: 2022-03-01

7.2 Systemadministration allgemein

7.2.1 Bootoptionen Cheatcodes

Zu Beginn des Bootvorgangs lässt sich die Kernel-Befehlszeile editieren, indem man, sobald das Grub-Menue erscheint, die Taste **e** drückt. Im Editiermodus navigiert man mit den Pfeiltasten zur Kernelzeile und fügt am Ende den oder die gewünschten Cheatcode ein. Als Trennzeichen dient das Leerzeichen. Der Bootvorgang wird mit der Tastenkombination **Strg**+**X** fortgesetzt.

Die nachstehenden Link führen zu der Handbuchseite mit den Tabellen für die Bootoptionen.

1. siduction spezifische Parameter (nur Live-CD)
2. Bootoptionen für den Grafikserver X
3. Allgemeine Parameter des Linux-Kernels
4. Werte für den allgemeinen Parameter **vga**

Ausführliche Referenzliste für Kernel-Bootcodes von kernel.org (Englisch, PDF)

7.2.2 systemd - Dienste verwalten

systemd kennt insgesamt 11 Unit-Typen. Die Units, mit denen wir im Alltag am häufigsten zu tun haben sind:

- systemd.service
- systemd.target
- systemd.device
- systemd.timer
- systemd.mount
- systemd.path

Einige der Unit-Typen stellen wir hier kurz vor. Ihre Namen geben bereits einen Hinweis auf die vorgesehene Funktionalität. Etwas ausführlichere Erläuterungen zu den Units beinhaltet unsere Handbuchseite Systemadministration.Systemd. Die

vollständige Dokumentation ist in den man-Pages `man systemd.unit`, `man systemd.special` und jeweils `man systemd.<Unit-Typ>` zu finden.

Mit dem Befehl

```
systemctl [OPTIONEN...] Befehl [UNIT...]
```

je nach den Units und den notwendigen Rechten als **user** oder **root** aufgerufen, wird das systemd-System gesteuert. *"systemctl"* kennt die Autovervollständigung mittels `TAB` und die Anzeige aller Variationen mittels `TAB TAB`. Bitte die man-Page `man systemctl` lesen.

Eine nach Typen sortierte Liste mit allen aktiven Units bzw. Unit-Dateien, geben die folgenden Befehle aus:

```
$ systemctl list-units          # für Units
$ systemctl list-unit-files     # für Unit-Dateien
```

mit der Option `-a` werden auch alle inaktiven Units bzw. Unit-Dateien ausgegeben.

7.2.3 systemd.service

Zum Starten oder Stoppen einer .service-Unit dienen die Befehle:

```
$ systemctl start <UNIT>.service
$ systemctl stop <UNIT>.service
$ systemctl restart <UNIT>.service
```

"Restart" ist z. B. nützlich, um dem Service eine geänderte Konfiguration bekannt zu geben. Sofern für die Aktion root-Rechte nötig sind, wird das root-Passwort abgefragt.
Zum Beenden eines Dienstes dient auch der Befehl:

```
$ systemctl kill -s SIGSTOP --kill-who=control <UNIT>.service
```

Mit *"kill"* stehen im Gegensatz zu *"stop"* die Optionen `-s, --signal=` und `--kill-who=` bereit. + *"-s"* sendet eines der Signale `SIGTERM`, `SIGINT`oder `SIGSTOP`. Vorgabe ist *"SIGTERM"*. + *"–kill-who="* erlaubt die Auswahl der Prozesse innerhalb der Hirarchie, an die ein Signal gesendet werden soll. Die Optionen sind `main`, `control` oder `all`. Damit wird dem Hauptprozess, den Kind-Prozesse

oder beiden das Signal gesendet. Vorgabe ist *“all”*.
Dieses Verhalten ähnelt dem altbekannten und weiterhin verwendbaren Befehl pkill, der weiter unten im Abschnitt Beenden eines Prozesses erläutert wird.

7.2.4 systemd - UNIT eingliedern

Damit eine (selbst erstellte) Unit beim Hochfahren des Rechners automatisch geladen wird, als **root**:

```
# systemctl enable <UNIT-Datei>
```

Dies erzeugt eine Gruppe von Symlinks entsprechend den Anforderungen in der Konfiguration der Unit. Im Anschluss wird automatisch die Systemverwalterkonfiguration neu geladen.

Der Befehl

```
# systemctl disable <UNIT-Datei>
```

entfernt die Symlinks wieder.

Beispiel
Wenn ein PC oder Laptop ohne Bluetooth Hardware im Einsatz ist, oder man kein Bluetooth verwenden möchte, entfernt der Befehl:

```
# systemctl disable bluetooth.service
```

die Symlinks aus allen Anforderungen und Abhängigkeiten innerhalb systemd und der Service ist nicht mehr verfügbar und wird auch nicht automatisch gestartet.

7.2.5 systemd-target ehemals Runlevel

Seit der Veröffentlichung von 2013.2 “December” benutzt siduction bereits systemd als Standard-Init-System.
Die alten sysvinit-Befehle werden weiterhin unterstützt. (hierzu ein Zitat aus `man systemd`: “… wird aus Kompatibilitätsgründen und da es leichter zu tippen ist, bereitgestellt.”)
Ausführlichere Informationen zum systemd enthält die Handbuchseite Systemadministration.systemd.

Die verschiedenen Runlevel, in die gebootet oder gewechselt wird, beschreibt systemd als **Ziel-Unit**. Sie besitzen die Erweiterung **.target**.

Ziel-Unit	Beschreibung
emergency.target	Startet in eine Notfall-Shell auf der Hauptkonsole. Es ist die minimalste Version eines Systemstarts, um eine interaktive Shell zu erlangen. Mit dieser Unit kann der Bootvorgang Schritt für Schritt begleitet werden.
rescue.target	Startet das Basissystem (einschließlich Systemeinhängungen) und eine Notfall-Shell. Im Vergleich zu multi-user.target könnte dieses Ziel als single-user.target betrachtet werden.
multi-user.target	Mehrbenutzersystem mit funktionierendem Netzwerk, ohne Grafikserver X. Diese Unit wird verwendet, wenn man X stoppen bzw. nicht in X booten möchte. Auf dieser Unit wird eine Systemaktualisierung (dist-upgrade) durchgeführt .
graphical.target	Die Unit für den Mehrbenutzermodus mit Netzwerkfähigkeit und einem laufenden X-Window-System.
default.target	Die Vorgabe-Unit, die systemd beim Systemstart startet. In siduction ist dies ein Symlink auf graphical.target (außer noX).

Ein Blick in die Dokumentation `man SYSTEMD.SPECIAL(7)` ist obligatorisch um die Zusammenhänge der verschiedenen *“xxx.target - Unit”* zu verstehen.

Um in den Runlevel zur Systemaktualisierung zu wechseln, ist im Terminal folgender Befehl als **root** zu verwenden:

```
# systemctl isolate multi-user.target
```

Wichtig ist hierbei der Befehl *“isolate”*, der dafür sorgt, dass alle Dienste und Services, welche die gewählte Unit nicht anfordert, beendet werden.

Um das System herunter zu fahren bzw. neu zu starten, sollte der Befehl

```
# systemctl poweroff
   bzw.
# systemctl reboot
```

verwendet werden. *“poweroff”* bzw. *“reboot”* (jeweils ohne .target) ist ein Befehl, der mehrere Unit in der richtigen Reihenfolge hereinholt, um das System geordnet zu beenden und ggf. einen Neustart auszuführen.

7.2.6 Beenden eines Prozesses

pgrep und pkill

Unabhängig von systemd ist `pgrep` und `pkill` ein sehr nützliches Duo um unliebsame Prozesse zu beenden. Mit Benutzer- oder root-Rechten in einer Konsole oder TTY ausgeführt:

```
$ pgreg <tab> <tab>
```

listet alle Prozesse mit ihrem Namen, aber ohne die Prozess-ID (PID) auf. Wir benutzen im Anschluss Firefox als Beispiel.
Die Option `-l` gibt die PID und den vollständigen Namen aus:

```
$ pgrep -l firefox
4279 firefox-esr
```

Um, sofern vorhanden, Unterprozesse anzuzeigen benutzen wir zusätzlich die Option `-P` und nur die PID:

```
$ pgrep -l -P 4279
4387 WebExtensions
4455 file:// Content
231999 Web Content
```

anschließend

```
$ pkill firefox-esr
```

beendet Firefox mit dem Standardsignal SIGTERM.
Mit der Option `--signal`, gefolgt von der Signalnummer oder dem Signalnamen, sendet pkill das gewünschte Signal an den Prozess. Eine übersichtliche Liste der Signale erhält man mit **`kill -L`**.

htop

Im Terminal eingegeben, ist htop eine gute Alternative, da sehr viele nützliche Informationen zu den Prozessen und zur Systemauslastung präsentiert werden. Dazu zählen eine Baumdarstellung, Filter- und Suchfunktion, Kill-Signal und einiges mehr. Die Bedienung ist selbsterklärend.

Notausgang

Als letzten Rettungsanker bevor der Netzstecker gezogen wird, kann man den Befehl **`killall -9`** im Terminal absetzen.

7.2.7 Vergessenes Rootpasswort

Ein vergessenes Rootpasswort kann nicht wiederhergestellt werden, aber ein neues kann gesetzt werden.

Dazu muss zuerst die Live-CD gebootet werden.

Als **root** muss die Rootpartition eingebunden werden (z. B. als /dev/sdb2)

```
mount /dev/sdb2 /media/sdb2
```

Nun folgen ein chroot in die Rootpartition (chroot=changed root … “veränderter Root”) und die Eingabe eines neuen Passwortes:

```
chroot /media/sdb2 passwd
```

7.2.8 Setzen neuer Passwörter

Um ein User-Passwort zu ändern, als **user** :

```
$ passwd
```

Um das Root-Passwort zu ändern, als **root** :

```
# passwd
```

Um ein User-Passwort als Administrator zu ändern, als **root** :

```
# passwd <user>
```

7.2.9 Schriftarten in siduction

Um, sofern nötig, die Darstellung der Schriften zu verbessern, ist es wichtig vorab die richtigen Einstellungen und Konfigurationen der Hardware zu prüfen.

Einstellungen prüfen

- Korrekte Grafiktreiber
  Einige neuere Grafikkarten von ATI und Nvidia harmonieren nicht besonders mit den freien Xorg-Treibern. Einzig vernünftige Lösung ist in diesen Fällen die Installation von proprietären, nicht quelloffenen Treibern. Aus rechtlichen Gründen kann siduction diese nicht vorinstallieren. Eine Anleitung zur Installation dieser Treiber findest Du auf der Seite Grafiktreiber des Handbuchs.

- Korrekte Bildschirmauflösungen und Bildwiederholungsraten
  Zuerst ist ein Blick in die technischen Unterlagen des Herstellers sinnvoll, entweder print oder online. Jeder Monitor hat seine eigene perfekte Einstellungskombination. Diese DCC-Werte werden in aller Regel richtig an das Betriebssystem übergeben. Nur manchmal muss manuell eingegriffen werden, um die Grundeinstellungen zu überschreiben.

  Um zu prüfen welche Einstellungen der X-Server zur Zeit verwendet, benutzen wir xrandr im Terminal:

```
$ xrandr
Screen 0: minimum 320 x 200, current 1680 x 1050, ⤢
   maximum 16384 x 16384
HDMI-1 disconnected (normal left inverted right x axis y⤢
    axis)
HDMI-2 connected 1680x1050+0+0 (normal left inverted ⤢
   right x axis y axis) 474mm x 296mm
  1680x1050     59.95*+
```

```
   1280x1024     75.02    60.02
   1440x900      59.90
   1024x768      75.03    60.00
   800x600       75.00    60.32
   640x480       75.00    59.94
   720x400       70.08
DP-1 disconnected (normal left inverted right x axis y ↙
    axis)
```

Der mit ***** markierte Wert kennzeichnet die verwendete Einstellung,
1680 x 1050 Pixel bei einer physikalischen Größe von 474 x 296 mm. Zusätzlich berechnen wir die tatsächliche Auflösung in Px/inch (dpi) um einen Anhaltspunkt für die Einstellungen der Schriften zu erhalten. Mit den oben ausgegebenen Werten erhalten wir 90 dpi.
1680 Px x 25,4 mm/inch : 474 mm = 90 Px/inch (dpi)

- Überprüfung
  Mit einem Zollstock oder Maßband ermitteln wir die tatsächliche Größe des Monitors. Das Ergebnis sollte um weniger als drei Millimeter von den durch xrandr ausgegebenen Werten abweichen.

Basiskonfiguration der Schriftarten

siduction nutzt freie Fonts, die sich in Debian als ausgewogen bewährt haben. In der graphischen Oberfläche kommen TTF- bzw. Outline-Schriften zur Anwendung. Wenn eigene Schriftarten gewählt werden, müssen eventuell neue Konfigurationsanpassungen vorgenommen werden, um das gewünschte Schriftbild zu erhalten.

Die systemweite Grundkonfiguration erfolgt im Terminal als **root** mittels:

```
# dpkg-reconfigure fontconfig-config
```

Bei den aufgerufenen Dialogen haben sich diese Einstellungen bewährt:

1. Bitte wählen Sie zur Bildschirmdarstellung die bevorzugte Methode zum Schriftabgleich (font tuning) aus.
   “autohinter”
2. Bitte wählen Sie, inwieweit Font-Hinting standardmäßig angewendet wird.
   “mittel”

3. Die Einbeziehung der Subpixel-Ebene verbessert die Textdarstellung auf Flachbildschirmen (LCD)
   “automatisch”
4. Standardmäßig nutzen Anwendungen, die fontconfig unterstützen, nur Outline-Schriften.
   Standardmäßig Bitmap-Schriften verwenden?
   “nein”

Anschließend ist

```
# dpkg-reconfigure fontconfig
```

notwendig um die Konfiguration neu zu schreiben.

Manchmal bedeutet der Neuaufbau des Font-Caches eine Lösung (der erste Befehl gilt der Datensicherung mit einem Datumsanhang, der zweite Befehl ist ohne Zeilenumbruch, d. h. in einer Zeile einzugeben):

```
# mv /etc/fonts/ /etc/fonts_$(date +%F)/
# apt-get install --reinstall --yes -o DPkg::Options::=--↙
   force-confmiss -o DPkg::Options::=--force-confnew ↙
   fontconfig fontconfig-config
```

7.2.10 Userkonfiguration

Darstellungsart, Größe, 4K-Display

Beachtet werden muss, dass jede Schriftart ein ideales Größenspektrum besitzt, sodass identische Größeneinstellungen nicht bei jeder Schriftart zu einem gleich guten Ergebnis führen muss.
Die Einstellungen kann man bequem in der graphischen Oberfläche vornehmen. Sie werden auf dem Desktop sofort wirksam, Anwendungen müssen zum Teil neu gestartet werden.
Die Liste zeigt, wo im Menü die Einstellungen zu finden sind.

- KDE Plasma
  Systemeinstellungen > *Schriftarten* > *Schriftarten*
  Systemeinstellungen > *Anzeige-Einrichtung* > *Anzeige-Einrichtung* > *Globale Skalierung*

- Gnome (Tweak Tool)
  Anwendungen > *Optimierungen* > *Schriften*
- XFCE
  Einstellungen > *Erscheinungsbild* > Reiter: *Schriften*

Begriffserklärung

"Kantenglättung / Antialising" :
Das ist die Helligkeitsabstufung der Nachbarpixel an den Kanten um bei Rundungen den Treppeneffekt zu vermindern. Es bewirkt aber eine gewisse Unschärfe der Schriftzeichen.

"Subpixel-Rendering / Farbreihenfolge / RGB" :
Das ist eine Erweiterung des Antialising für LCD-Bildschirme, indem zusätzlich die Farbkomponenten eines Pixels angesteuert werden.

"Hinting" :
Ist die Anpassung (Veränderung) der Schriftzeichen an das Pixelrasters des Bildschirms. Dadurch verringert sich der Bedarf an Antialising, aber die Schriftform entspricht nicht mehr genau den Vorgaben, es sei denn, die Entwickler der Schrift haben bereits Hintingvarianten integriert. Bei **4K**-Bildschirmen ist Hinting meist nicht notwendig.

"DPI-Wert / Skalierungsfaktor" :
Die Einstellmöglichkeit eines anderen DPI-Wertes bzw. einer anderen Größe nur für die Schriften. Hier lässt sich die Darstellung auf einem **4K**-Bildschirm schnell verbessern. Die folgende Tabelle verdeutlicht den Zusammenhang zwischen der Bildschirmdiagonalen und dem DPI-Wert bei **4k**-Bildschirmen.

4k Auflösung: 3840 x 2160 (16:9)

Diagonale	X-Achse	Y-Achse	DPI
24 Zoll	531 mm	299 mm | 18	184
27 Zoll	598 mm	336 mm | 16	163
28 Zoll	620 mm	349 mm | 15	157
32 Zoll	708 mm	398 mm | 13	138
37 Zoll	819 mm	461 mm | 11	119
42 Zoll	930 mm	523 mm | 10	105

Demnach ist bei 4k-Bildschirmen mit 24 Zoll Diagonale ein Skalierungsfaktor von 2,0 und mit 37 Zoll Diagonale ein Skalierungsfaktor von 1,2 erforderlich um etwa gleiche Darstellungen entsprechend SXGA oder WSXGA Bildschirmen mit 90 DPI zu erhalten.

7.2.11 CUPS - das Drucksystem

KDE hat einen großen Abschnitt zu CUPS in der KDE-Hilfe. Trotzdem folgt nun eine Anleitung, was man bei Problemen mit CUPS nach einem full-upgrade tun kann. Eine der bekannten Lösungen ist:

```
# modprobe lp
# echo lp >> /etc/modules
# apt purge cups
# apt install cups
        ODER
# apt install cups printer-driver-gutenprint hplip
```

CUPS wird nun neu gestartet:

```
# systemctl restart cups.service
```

Im Anschluss daran wird ein Web-Browser geöffnet und in die Adresszeile eingegeben:

http://localhost:631

Ein kleines Problem tritt auf, wenn CUPS zur Legitimation die entsprechende Dialog-Box öffnet. Dort ist gelegentlich der eigene Benutzername bereits eingetragen und das Passwort wird erwartet. Die Eingabe des Benutzerpassworts ist jedoch nicht zielführend. Es geht nichts. Die Lösung ist, den Benutzernamen in **root** zu ändern und das root-Passwort einzugeben.

Die OpenPrinting-Datenbank beinhaltet umfangreiche Informationen über verschiedenste Drucker und deren Treiber. Es stehen Treiber, Spezifikationen und Konfigurations-Tools zur Verfügung. Die Firma Samsung lieferte früher eigene Linux-Treiber für ihre Drucker. Nach dem Verkauf der Druckersparte an HP war die Downloadseite nicht mehr erreichbar und HP nahm die Samsung-Treiber leider nicht in die *“hplib”* auf. Derzeit funktioniert für Samsung-Drucker und

Samsung-Multifunktionsgeräte am ehesten das Paket `printer-driver-splix` . CUPS ist gerade im Umbruch und geht in Richtung Drucken ohne Treiber per IPP-Everywhere.

7.2.12 Sound in siduction

In älteren siduction Installationen ist der Ton in der Grundeinstellung deaktiviert.

Die meisten Tonprobleme lassen sich lösen, indem man auf das Sound-Ikon in der Kontrollleiste klickt, den Mischer öffnet und das Häkchen von “stumm” oder “mute” entfernt bzw. den entsprechenden Schieber betätigt. Ist das Lautsprecher-Symbol nicht vorhanden, genügt ein Rechtsklick auf die Kontrollleiste, dann die Auswahl

in KDE: *Kontrollleiste Optionen > Miniprogramme hinzufügen…*
in XFCE: *Leiste > Neue Elemente hinzufügen…*

und das gewünschte Modul auswählen.

KDE Plasma

Ein Rechtsklick auf das Lautsprechersymbol in der Kontrollleiste öffnet das Einstellungsfenster für die Soundausgabe. Die Benutzerführung ist selbsterklärend.

GNOME

Ein Rechtsklick auf das Lautsprechersymbol in der Kontrollleiste öffnet ein Dropdown-Menü, dass einen Schieber für die Lautstärke enthält.
Weitere Einstellungen sind wie folgt möglich:

Rechtsklick auf die Arbeitsfläche > *Einstellungen* > *Audio*

XFCE Pulse-Audio

Die Einstellungen erfolgen über das Lautsprechersymbol (Puls-Audio-Modul) in der Kontrollleiste. Auch hier ist die Benutzerführung selbsterklärend. Fehlt das Symbol, kann man sich auf die Schnelle mit einem Terminal und dem Befehl

```
$ pavucontrol
```

behelfen und nimmt im neu geöffneten Fenster die Einstellungen vor.

Alsamixer

Wer alsamixer bevorzugt, findet diesen im Paket alsa-utils:

```
# apt update
# apt install alsa-utils
# exit
```

Die gewünschten Sound-Einstellungen werden als **<user>** von einem Terminal aus vorgenommen:

```
$ alsamixer
```

Zuletzt bearbeitet: 2021-11-29

7.3 Doas - Alternative zu Sudo

Wir, das siduction Team, haben uns für einen echten Root-Account entschieden und Sudo nicht konfiguriert. Für Benutzer die Sudo gewohnt sind und nicht auf seine Funktionalität verzichten möchten, bietet sich die schlanke Alternative Doas an. Doas ist, im Vergleich zu Sudo mit nur etwa 1/100 an Codezeilen, auf Desktop-Systeme zugeschnitten. Mit *siduction 2021.3 wintersky* wird Doas in Version 6.8.1-3 automatisch installiert, ist jedoch noch nicht vollständig konfiguriert.

7.3.1 Doas konfigurieren

Um Doas benutzen zu können fehlt einzig die Konfigurationsdatei `/etc/doas.conf`. Sie enthält zeilenweise Regeln, die Aktionen einem Benutzer zuweisen. Eine *"#"* leitet Kommentare ein. Die Zeilen liest Doas nacheinander, wobei die Aktion der letzten zutreffenden Regel ausgeführt wird. Für das Verständnis der Regeln in der Konfigurationsdatei sind einige Dinge zu beachten.
- Nur Aktionen, für die mindestens eine Regel zutrifft, werden ausgeführt. - Dadurch, dass Doas die Regeln zeilenweise nacheinander auswertet, lassen sich Hierarchien aufbauen. - Bei Regeln die Kommandos mit Argumenten enthalten, sind die Argumente exakt und vollständig anzugeben. - Regeln mit Kommandos die variable Argumente benötigen sind nicht möglich. - Doas prüft vor der Ausführung der angeforderten Aktion die Syntax der Konfigurationsdatei. Bei fehlerhaften Regeln erfolgt die Ausgabe `doas: syntax error at line 4` und Doas beendet sich. Der schreibende Zugriff auf die Konfigurationsdatei ist dann nur mit dem Root-Account möglich.

Besonders einfach ist die Konfiguration, wenn auf dem siduction System nur ein User-Account existiert. Eine einzige Zeile reicht um mittels vorangestelltem "doas" Befehle mit root-Rechten auszuführen.
Melde dich in einem Terminal als **root** an und führe folgenden Befehl aus, wobei "tux" durch den Namen deines Benutzer-Account zu ersetzen ist.

```
tux@sidu:~$ su
Passwort:
root@sidu:/home/tux# echo "permit keepenv nopass tux" > /etc/
   doas.conf
root@sidu:/home/tux# exit
tux@sidu:~$
```

Die Konfigurationszeile setzt sich zusammen aus:
Der Aktion *“permit|deny”* (erlauben|verbieten) mit
der Option *“keepenv”* (Umgebungsvariablen beibehalten - ermöglicht das Starten von graphischen Programmen wie gparted),
der Option *“nopass|persist”* (keine Passwortabfrage | die einmalige Passwort Eingabe bleibt einen begrenzten Zeitraum gültig) und
dem Benutzer *“tux”*, auf den die Aktion anzuwenden ist.

Steht der Benutzername für sich allein, so darf **tux** Befehle als beliebiger, auf dem System vorhandener Benutzer ausführen. Die Vorgabe ist **root**. Soll die Ausführung der Aktion nur mit den Rechten eines anderen Benutzers als root erlaubt sein, ist der Name innerhalb der Regel anzugeben (z.B. *“tux as anne”*). Statt des Benutzers kann durch das Voranstellen eines `:` eine Gruppe (z.B. *“:vboxusers”*) Berechtigungen erlangen.

7.3.2 Doas und mehrere Benutzer

Beispiel
Auf dem Arbeitsplatz-PC dürfen sich außer **tux** drei weitere Benutzer mit den Namen **anne**, **bob** und **lisa** anmelden.
Anne möchte nur Bob erlauben zwei ihrer Skripte aus ihrem Verzeichnis `/home/anne/bin` auszuführen. Die Rechte an ihren Skripten hat Anne restriktiv auf `700` gesetzt.
Lisa ist besonders vertrauenswürdig, weshalb sie für die Systemupgrades zuständig sein soll.

Wir nutzen als **tux** sogleich Doas in einem Terminal um die Konfigurationsdatei zu bearbeiten.

```
tux@sidu:~$ doas mcedit /etc/doas.conf
```

Die zuvor genannten Berechtigungen setzen wir in Regeln um und ergänzen die Datei um einige Kommentare.

```
# doas config file /etc/doas.conf

# tux erhält root-Rechte
permit keepenv nopass tux
```

```
# bob darf script von anne ausführen
permit bob as anne cmd /home/anne/bin/script1 args -n
permit bob as anne cmd /home/anne/bin/script2 args

# lisa darf Systemupgrade ausführen
permit persist lisa cmd init
deny lisa cmd init args 1
deny lisa cmd init args 5
permit persist lisa cmd apt args update
permit persist lisa cmd apt args full-upgrade
```

Erklärungen

Bob darf die Skripte *“script1”* und *“script2”* innerhalb Annes `/home/anne/bin` Verzeichnis ausführen. Ersteres ausschließlich mit der Argument `-n`, dem Zweiten darf kein Argument mitgegeben werden. Die Angabe von *“args”* in der Regelzeile für das script2 ohne ein folgendes Argument, erzwingt den Aufruf der Datei ohne Argument und damit ohne möglicherweise schädlichen Code. Bob muss dem Aufruf der Skripte mit der Option `-u` den Benutzernamen mitgeben.

```
bob@sidu:~$ doas -u anne /home/anne/bin/script1 -n
doas (bob@sidu) password:

bob@sidu:~$
```

Das Skript wurde nach Eingabe des Benutzerpasswortes von Bob ohne Kommentar ausgeführt.

Damit **Lisa** das Systemupgrade ausführen kann, soll sie zum *multi-user.target* (init 3) wechseln und nach Abschluss einen *systemctl reboot* (init 6) durchführen. Die Regelzeile *“permit persist lisa as root cmd init”* ohne die Angabe von args bewirkt, dass alle Aufrufe von init erlaubt sind, außer denen, die in den nachfolgenden Regeln verboten werden. Deshalb kann sie nicht direkt vom *multi-user.target* in das *graphical.target* (init 5) wechseln. Hier sehen wir den Aufbau einer Hierarchie.

Hinweise

Wer immer wieder `sudo` eintippt, dem hilft die Zeile `alias sudo="doas"` in seiner `.bashrc`.

Doas spielt seinen entscheidenden Vorteil dort aus, wo nur einem User mittels

doas root-Rechte erteilt werden. Das obige Beispiel mit Lisa zeigt, wie umfangreich die Konfiguration für eine eingeschränkte Rechtevergabe werden kann. Außerdem ist eine Regel für einen Programmaufruf mit variablen Argumenten (z.B. `apt ↙ install <Paketname>`) nicht möglich.

Quellen

man doas

man doas.conf

github, doas

LinuxNews, Linux Rechtemanagement, sudo durch doas ersetzen

LinuxUser 08.2021, Kleiner Bruder

Zuletzt bearbeitet: 2022-04-01

7.4 APT Paketverwaltung

APT ist eine Abkürzung für **A**dvanced **P**ackaging **T**ool und stellt eine Sammlung von Programmen und Skripten bereit, welche das System und den Administrator bei der Installation und Verwaltung von Debian-Paketen unterstützt.
Eine vollständige Beschreibung des APT-Systems findet man in Debians APT-HOWTO

7.4.1 apt und apt-get

apt ist als Endanwenderschnittstelle gedacht und aktiviert verglichen mit spezialisierteren Werkzeugen wie apt-get und apt-cache standardmäßig einige für den interaktiven Gebrauch besser geeignete Optionen. Mit apt stehen nicht alle Optionen von apt-get und apt-cache zur Verfügung. Bitte die man-Pages von apt, apt-get und apt-cache lesen. Die folgende Tabelle zeigt die jeweiligen Befehle und ihre grundlegende Bedeutung.

apt	apt-get	Kurzinfo
apt update	apt-get update	Auffrischen der Paketdatenbank.
apt upgrade	apt-get upgrade	Aktualisiert das System auf die neuesten, zur Verfügung stehenden Paketversionen.
apt full-upgrade	apt-get dist-upgrade	Aktualisiert das System auf die neuesten, zur Verfügung stehenden Paketversionen auch wenn dadurch bereits installierte Pakete entfernt werden müssen.

apt	apt-get	Kurzinfo
apt full-upgrade -d	apt-get dist-upgrade -d	Aktualisierung das System wie zuvor, jedoch wird nur der Download durchgeführt und nichts installiert.
apt install	apt-get install	Installieren eines oder mehrerer Pakete.
apt remove	apt-get remove	Entfernen eines oder mehrerer Pakete.
apt purge	apt-get purge	Entfernen eines oder mehrerer Pakete incl. der Konfigurationsdateien.
-	apt-mark hold	Verhindert, dass apt eine andere Version das Paketes installiert.
-	apt-mark unhold	Hebt den Befehl ‘apt-mark hold’ auf.
apt search	apt-get search	Sucht entsprechend des eingegebenen Musters nach Paketen. (regex möglich)
apt show	apt-cache show	Anzeige der Details eines Paketes.
apt list	apt-cache policy	Zeigt die installierte, oder installierbare Version eines Paketes.

7.4.2 sources.list - Liste der Quellen

Das APT-System benötigt eine Konfigurationsdatei, welche Informationen über den Ort der installierbaren und aktualisierbaren Pakete beinhaltet. Im allgemeinen nennt man diese Datei sources.list. Moderne Systeme benutzen mittlerweile modularisierte Sourcen um die Übersicht zu verbessern.

siduction stellt die Quellen in diesem Ordner bereit:

`/etc/apt/sources.list.d/`

Innerhalb dieses Verzeichnisses befinden sich standardmäßig folgende Dateien:

`debian.list`
`extra.list`
`fixes.list`

Dies hat den Vorteil, dass leichter automatisch aus Spiegelservern gewählt werden kann (“mirror switching”), und auch das Ergänzen oder Austauschen von Quellen-Listen ist so einfacher zu gestalten.

Eigene Quellen-Listen-Dateien können mit der Benennung `/etc/apt/sources.list.d/xxxx.list` hinzugefügt werden. Auf einem siduction könnte `/etc/apt/sources.list.d/extra.list` zum Beispiel so aussehen:

```
This is the default mirror, choosen at first boot.
# One might consider to choose the geographical nearest or
   the fastest mirror.
deb     http://packages.siduction.org/extra unstable main
   contrib non-free
#deb-src http://packages.siduction.org/extra unstable main
   contrib non-free
```

unter `/etc/apt/sources.list.d/fixes.list` könnte es so aussehen:

```
deb      https://packages.siduction.org/fixes unstable main
   contrib non-free
#deb-src https://packages.siduction.org/fixes unstable main
   contrib non-free
```

und `/etc/apt/sources.list.d/debian.list` enthält dann das eigentliche Debian Repo:

```
# debian loadbalancer
deb     http://deb.debian.org/debian/ unstable main contrib
   non-free
#deb-src http://deb.debian.org/debian/ unstable main contrib
   non-free
```

Weitere Einträge für optionale siduction Repositories finden sich auf siduction Repositories.

Fügt man zum Beispiel ein oder mehrere Debian Repositories hinzu, so würde dies folgender maßen aussehen:

```
#Debian
# Unstable
deb http://ftp.us.debian.org/debian/ unstable main contrib ↙
   non-free
#deb-src http://ftp.us.debian.org/debian/ unstable main ↙
   contrib non-free

# Testing
#deb http://ftp.us.debian.org/debian/ testing main contrib ↙
   non-free
#deb-src http://ftp.us.debian.org/debian/ testing main ↙
   contrib non-free

# Experimental
#deb http://ftp.us.debian.org/debian/ experimental main ↙
   contrib non-free
#deb-src http://ftp.us.debian.org/debian/ experimental main ↙
   contrib non-free
```

ZUR BEACHTUNG:
In diesem Beispiel wird der US-amerikanische Debian-Spiegelserver beginnend mit ftp.us verwendet. Diese Einstellung kann als root geändert werden, indem der Landes-Code angepasst wird (zum Beispiel: ftp.at, ftp.de). Die meisten Länder haben lokale Debian-Spiegelserver zur Verfügung. Dies bietet für den Anwender eine höhere Anbindungsgeschwindigkeit und spart auch Bandbreite.

Liste der aktuell verfügbaren Debian-Server und deren Spiegelserver.

7.4.3 apt update

Um aktualisierte Informationen über die Pakete zu erhalten, wird eine Datenbank mit den benötigten Einträgen vorgehalten. Das Programm apt benutzt sie bei der Installation eines Pakets, um alle Abhängigkeiten aufzulösen und somit zu garan-

tieren, dass die ausgewählten Pakete funktionieren. Die Erstellung bzw. Aktualisierung dieser Datenbank wird mit dem Befehl **`apt update`** durchgeführt.

```
root@siduction# apt update
Holen:1 http://siduction.org sid Release.gpg [189B]
Holen:2 http://siduction.org sid Release.gpg [189B]
Holen:3 http://siduction.org sid Release.gpg [189B]
Holen:4 http://ftp.de.debian.org unstable Release.gpg [189B]
Holen:5 http://siduction.org sid Release [34.1kB]
Holen:6 http://ftp.de.debian.org unstable Release [79.6kB]
Es wurden 404 kB in 8 s geholt (50,8 kB/s).
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut.
Statusinformationen werden eingelesen.... Fertig
Aktualisierung für 48 Pakete verfügbar. Führen Sie »apt list ↙
   --upgradable« aus, um sie anzuzeigen.
```

7.4.4 Pakete installieren

Ist uns der Name des Pakets bekannt, reicht der Befehl **`apt install <Paketname>`**.
(Weiter unten wird gezeigt, wie man ein Paket finden kann.)

> **Warnhinweis:**
> Pakete, die nicht im multi-user.target (ehemals Runlevel 3) installiert werden, können große, nicht unterstützbare Probleme mit sich bringen!

Deshalb empfehlen wir folgenden Ablauf:

1. Aus der Desktopumgebung abmelden.
2. Mit **`Ctrl`**+**`Alt`**+**`F2`** auf die Textkonsole wechseln.
3. Einloggen als **root**.

Anschließend das gewünschte Programmpaket installieren:

```
init 3
apt update
apt install <Paketname>
init 5 && exit
```

Im unteren Beispiel wird das Paket “funtools” installiert.

```
root@siduction# apt install funtools
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut.
Statusinformationen werden eingelesen.... Fertig
Die folgenden zusätzlichen Pakete werden installiert:
  libfuntools1 libwcstools1
Die folgenden NEUEN Pakete werden installiert:
  funtools libfuntools1 libwcstools1
0 aktualisiert, 3 neu installiert, 0 zu entfernen und 48 ⇙
   nicht aktualisiert.
Es müssen 739 kB an Archiven heruntergeladen werden.
Nach dieser Operation werden 2.083 kB Plattenplatz zusätzlich⇙
    benutzt.
Möchten Sie fortfahren? [J/n] j
Holen:1 http://deb.debian.org/debian unstable/main amd64 ⇙
   libwcstools1 amd64 3.9.5-3 [331 kB]
Holen:2 http://deb.debian.org/debian unstable/main amd64 ⇙
   libfuntools1 amd64 1.4.7-4 [231 kB]
Holen:3 http://deb.debian.org/debian unstable/main amd64 ⇙
   funtools amd64 1.4.7-4 [177 kB]
Es wurden 739 kB in 0 s geholt (1.678 kB/s).
Vormals nicht ausgewähltes Paket libwcstools1:amd64 wird ⇙
   gewählt.
(Lese Datenbank ... 279741 Dateien und Verzeichnisse sind ⇙
   derzeit installiert.)
Vorbereitung zum Entpacken von .../libwcstools1_3.9.5-3_amd64⇙
   .deb ...
Entpacken von libwcstools1:amd64 (3.9.5-3) ...
Vormals nicht ausgewähltes Paket libfuntools1:amd64 wird ⇙
   gewählt.
Vorbereitung zum Entpacken von .../libfuntools1_1.4.7-4_amd64⇙
   .deb ...
Entpacken von libfuntools1:amd64 (1.4.7-4) ...
Vormals nicht ausgewähltes Paket funtools wird gewählt.
Vorbereitung zum Entpacken von .../funtools_1.4.7-4_amd64.deb⇙
    ...
Entpacken von funtools (1.4.7-4) ...
```

```
libwcstools1:amd64 (3.9.5-3) wird eingerichtet ...
libfuntools1:amd64 (1.4.7-4) wird eingerichtet ...
funtools (1.4.7-4) wird eingerichtet ...
Trigger für man-db (2.8.5-2) werden verarbeitet ...
Trigger für libc-bin (2.28-8) werden verarbeitet ...
```

7.4.5 Pakete entfernen

Der Befehl **`apt remove <Paketname>`** entfernt ein Paket. Abhängigkeiten werden dabei nicht entfernt:

```
root@siduction# apt remove gaim
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut.
Statusinformationen werden eingelesen.... Fertig
Die folgenden Pakete wurden automatisch installiert und ⤢
   werden nicht mehr benötigt:
     libfuntools1 libwcstools1
Verwenden Sie »sudo apt autoremove«, um sie zu entfernen.
Die folgenden Pakete werden ENTFERNT:
     funtools
0 aktualisiert, 0 neu installiert, 1 zu entfernen und 48 ⤢
   nicht aktualisiert.
Nach dieser Operation werden 505 kB Plattenplatz freigegeben.
Möchten Sie fortfahren? [J/n] j
(Lese Datenbank ... 279786 Dateien und Verzeichnisse sind ⤢
   derzeit installiert.)
Entfernen von funtools (1.4.7-4) ...
Trigger für man-db (2.8.5-2) werden verarbeitet ...
```

Im letzten Fall werden die Konfigurationsdateien nicht vom System entfernt, sie können bei einer späteren Neuinstallation des Programmpakets (im Beispielfall gaim) wieder verwendet werden. Sollen auch die Konfigurationsdateien entfernt werden, setzen wir den Befehl **`apt purge funtools`** ab.

So werden auch die Konfigurationsdateien mit entfernt. Will man sehen, ob Konfigurationsdateien von bereits entfernten Programmen noch auf dem System verblieben sind, kommt man mit `dpkg` ganz einfach zu einem Ergebnis:

```
dpkg -l | grep ^rc
rc  colord             1.4.3-3.1       amd64  system service ⤶
   to manage device colour profiles -- system daemon
rc  hplip              3.18.10+dfsg0-1 amd64  HP Linux ⤶
   Printing and Imaging System (HPLIP)
rc  libsensors4:amd64  1:3.4.0-4       amd64  library to read⤶
    temperature/voltage/fan sensors
rc  sane               1.0.14-13.1     amd64  scanner ⤶
   graphical frontends
rc  sane-utils         1.0.27-3.1      amd64  API library for⤶
    scanners -- utilities
rc  systemd-coredump   240-1           amd64  tools for ⤶
   storing and retrieving coredumps
```

Die hier gelisteten Pakete wurden removed, ohne purgen.

7.4.6 Hold oder Downgraden eines Pakets

Manchmal kann es notwendig sein, auf eine frühere Version eines Pakets zurückzugreifen, da die neueste Version einen gravierenden Fehler aufweist.

Hold (Halten)

apt bietet mit `apt-mark` die Möglichkeit verschiedene Einstellungen für ein Paket vorzunehmen. Die Option `hold` schützt das Paket vor Änderungen durch apt.

```
apt-mark hold <paket>
```

So beendet man den Hold eines Pakets:

```
apt-mark unhold <paket>
```

So sucht man nach Paketen, die auf Hold gesetzt sind:

```
apt-mark showhold
```

Bitte bedenkt, dass hold nur eine Notfallmaßnahme ist. Man wird sich Probleme einhandeln, wenn man vergisst, einen hold wieder zeitnah aufzuheben. Das gilt

umso mehr, je mehr (essentielle) Abhängigkeiten das Paket hat. Also: holds bitte nur im Notfall und schnellstmöglich wieder aufheben.

Downgraden (Deaktualisierung)

Debian unterstützt keinen Downgrade von Paketen. In einfachen Fällen kann das Installieren älterer Versionen gelingen, es kann aber auch spektakulär fehlschlagen. Mehr Informationen im englischsprachigen Debian-Handbuch unter dem Kapitel Emergency downgrading.

Obwohl ein Downgrade nicht unterstützt ist, kann er bei einfachen Paketen gelingen. Die Schritte für einen Downgrade werden nun am Paket kmahjongg demonstriert:

Die Quellen von Unstable werden in `/etc/apt/sources.list.d/debian.list` mit einem Rautezeichen “#” versehen und die Quellen für Testing hinzugefügt. Danach führen wir die folgenden Befehle aus:

```
apt update
apt install kmahjongg/testing
```

Das nun installierte Paket wird vor Aktualisierungen geschützt, auf Hold gesetzt:

```
apt-mark hold kmahjongg
```

Anschließend machen wir die Änderungen in `/etc/apt/sources.list.d/debian.list` wieder rückgängig. Also die Quellen für Testing mit einem Rautezeichen “#” versehen, während die Rautezeichen vor den Quellen für Unstable wieder entfernt werden. Nach dem Speichern der Änderungen:

```
apt update
```

Wenn ein neues, fehlerfreies Paket in sid eintrifft, kann man die neueste Version wieder installieren, nachden man den “hold”-Status beendet hat:

```
apt-mark unhold kmahjongg
apt update
apt install kmahjongg
    oder
apt full-upgrade
```

7.4.7 Aktualisierung des Systems

Eine Aktualisierung des ganzen Systems wird mit diesem Befehl durchgeführt: **`apt full-upgrade`**. Vor einer solchen Maßnahme sollten die aktuellen Upgradewarnungen auf der Hauptseite von siduction beachtet werden, um zu prüfen, ob Pakete des eigenen Systems betroffen sind. Wenn ein installiertes Paket behalten, also auf hold gesetzt werden sollte, verweisen wir auf den Abschnitt Downgrade bzw. Hold eines Pakets.

Ein einfaches **`apt upgrade`** von Debian Sid ist normalerweise nicht empfohlen. Es kann aber hilfreich sein, wenn eine Situation mit vielen gehaltenen oder zu entfernenden Paketen vorliegt, um von der Situation nicht betroffene Pakete zu aktualisieren.

Wie regelmäßig soll eine Systemaktualisierung durchgeführt werden?
Eine Systemaktualisierung soll regelmäßig durchgeführt werden, alle ein bis zwei Wochen haben sich als guter Richtwert erwiesen. Auch bei monatlichen Systemaktualisierungen sollte es zu keinen nennenswerten Problemen kommen. Theoretisch kann das System mehrmals täglich nach der Synchronisation der Spiegelserver alle 6 Stunden aktualisiert werden.

Die Erfahrungen zeigen, dass länger als zwei, maximal drei Monate nicht gewartet werden sollte. Besonders beachtet sollten Programmpakete werden, welche nicht aus den siduction- oder Debian-Repositorien stammen oder selbst kompiliert wurden, da diese nach einer Systemaktualisierung mittels full-upgrade wegen Inkompatibilitäten ihre Funktionsfähigkeit verlieren können.

Aktualisierung nicht mit Live-Medium

Die Möglichkeit der Aktualisierung einer siduction-Installation mittels eines Live-Mediums existiert nicht. Weiter unten beschreiben wir ausführlich den Aktualisierungsvorgang und warum apt verwendet werden sollte.

7.4.8 Aktualisierbare Pakete

Nachdem die interne Datenbank aktualisiert wurde, kann man herausfinden, für welche Pakete eine neuere Version existiert (zuerst muss apt-show-versions installiert werden):

```
root@siduction# apt-show-versions -u
```

```
libpam-runtime/unstable upgradeable from 0.79-1 to 0.79-3
passwd/unstable upgradeable from 1:4.0.12-5 to 1:4.0.12-6
teclasat/unstable upgradeable from 0.7m02-1 to 0.7n01-1
libpam-modules/unstable upgradeable from 0.79-1 to ⤶
   0.79-3.........
```

Das gleiche erreicht man mit:

```
apt list --upgradable
```

Die Aktualisierung eines einzelnes Pakets (hier z. B. debtags-1.6.6.0) kann unter Berücksichtigung der Abhängigkeiten vorgenommen werden mit:

```
root@siduction# apt install debtags-1.6.6.0
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut... Fertig
Die folgenden Pakete werden ENTFERNT:
  apt-index-watcher
Die folgenden Pakete werden aktualisiert:
  debtags
1 aktualisiert, 0 neu installiert, 1 zu entfernen und 0 nicht⤶
    aktualisiert.
Es müssen 660kB Archive geholt werden.
Nach dem Auspacken werden 1991kB Plattenplatz freigegeben ⤶
   worden sein.
Möchtest Du fortfahren [J/n]?
Hole:1 http://ftp.de.debian.org unstable/main debtags 1.6.6 ⤶
   [660kB]
Es wurden 660kB in 1s geholt (513kB/s)
(Lese Datenbank ... 138695 Dateien und Verzeichnisse sind ⤶
   derzeit installiert.)
Entferne apt-index-watcher ...
(Lese Datenbank ... 138692 Dateien und Verzeichnisse sind ⤶
   derzeit installiert.)
Vorbereiten zum Ersetzen von debtags 1.6.2 (durch .../⤶
   debtags_1.6.6_i386.deb) ...
Entpacke Ersatz für debtags ...
Richte debtags ein (1.6.6) ...
```

```
Installiere neue Version der Konfigurationsdatei /etc/debtags↙
   /sources.list ...
```

(Nur) Downloaden

Eine wenig bekannte, aber großartige Möglichkeit ist die Option -d:

```
apt update && apt full-upgrade -d
```

`-d` ermöglicht, die Pakete eines full-upgrades lokal zu speichern, ohne dass sie installiert werden. Dies kann in einer Konsole durchgeführt werden, während man in X ist. Der full-upgrade selbst kann zu einem späteren Zeitpunkt in init 3 erfolgen. Dadurch erhält man auch die Möglichkeit, nach eventuellen Warnungen zu recherchieren und danach zu entscheiden, ob man die Aktualisierung durchführen möchte oder nicht:

```
root@siduction#apt full-upgrade -d
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following NEW packages will be installed:
  elinks-data
The following packages have been kept back:
  git-core git-gui git-svn gitk icedove libmpich1.0ldbl
The following packages will be upgraded:
  alsa-base bsdutils ceni configure-ndiswrapper debhelper
  discover1-data elinks file fuse-utils gnucash.........
35 upgraded, 1 newly installed, 0 to remove and 6 not ↙
   upgraded.
Need to get 23.4MB of archives.
After this operation, 594kB of additional disk space will be ↙
   used.
Möchtest Du fortfahren [J/n]?J
```

J lädt die zu aktualisierenden bzw. neu zu installierenden Pakete, ohne das installierte System zu verändern.

Nach dem Download der Pakete mittels `full-upgrade -d` können diese jederzeit entsprechend dem Vorgehen im folgendem Absatz installiert werden.

7.4.9 full-upgrade ausführen

> **Warnhinweis:**
> Eine Systemaktualisierung, die nicht im multi-user.target (ehemals Runlevel 3) durchgeführt wird, kann zu Problemen führen, wenn es um Updates der installierten Desktop-Umgebung oder des X-Servers geht!

Besuche vor einer Systemaktualisierung die siduction-Homepage, um eventuelle Upgradewarnungen in Erfahrung zu bringen. Diese Warnungen sind wegen der Struktur von Debian sid/unstable notwendig, welches mehrmals täglich neue Programmpakete in seine Repositorien aufnimmt.

Zu beachten ist der folgende Ablauf:

1. Aus der Desktopumgebung abmelden.
   (Diese Vorgehensweise wird heutzutage nur noch bei der Aktualisierung von X oder der Desktop-Umgebung selbst empfohlen, schadet aber auch in anderen Fällen nicht.)
2. Mit **Ctrl**+**Alt**+**F2** auf die Textkonsole wechseln.
3. Einloggen als **root**.

Anschließend folgende Befehle ausführen:

```
init 3
apt update
apt full-upgrade
apt clean
init 5 && exit
```

Wurde ein neuer Kernel installiert, ist an Stelle von *"init 5"* der Befehl **systemctl reboot** oder **init 6** notwendig, um in den neuen Kernel zu booten.

7.4.10 Warum ausschließlich apt verwenden

Zum Installieren, Löschen und Durchführen einer Systemaktualisierung soll *apt* verwendet werden.
Bitte von Systemaktualisierungen mit Anwendungen wie `synaptic`, `adept` oder `kpackage` absehen!

Die genannten Programme sind exzellent für eine Installation von *Debian stable* und sie eignen sich sehr gut dazu Programmpakete zu suchen, aber sie sind nicht angepasst an die besonderen Aufgaben der dynamischen Distribution Debian Sid. Sie können nicht immer die umfassenden Änderungen in Sid (Änderungen von Abhängigkeiten, Benennungskonventionen, Skripten u.a.) korrekt auflösen. Es handelt sich dabei nicht um Fehler in diesen Programmen oder Fehler der Entwickler.

Paketmanager wie adept, synaptic und kpackage sind - technisch gesprochen - nicht deterministisch. Bei Verwendung einer dynamischen Distribution wie Debian Sid unter Hinzunahme von Drittrepositorien, deren Qualität nicht vom Debian-Team getestet sein kann, kann eine Systemaktualisierung zur Katastrophe führen, da diese Paketmanager durch automatische Lösungsversuche falsche Entscheidungen treffen können.

Weiterhin ist zu beachten, dass alle GUI-Paketmanager in X ausgeführt werden müssen. Systemaktualisierungen in X (selbst ein ohnehin nicht empfohlenes ‘apt upgrade’) werden früher oder später dazu führen, dass man sein System irreversibel beschädigt.

Im Gegensatz dazu führt apt ausschließlich das durch, was angefragt ist. Bei unvollständigen Abhängigkeiten in Sid, sprich: wenn das System bricht (dies kann in Sid bei Strukturänderungen vorkommen), können die Ursachen genau festgestellt und dadurch repariert oder umgangen werden. Das eigene System “bricht” nicht. Falls also eine Systemaktualisierung dem Gefühl nach das halbe System löschen möchte, überlässt apt dem Administrator die Entscheidung, was zu tun ist, und handelt nicht eigenmächtig.

Dies ist der Grund, warum Debian-Builds apt nutzen und nicht andere Paketmanager.

7.4.11 Programmpakete suchen

Das APT-System bietet eine Reihe nützlicher Suchbefehle, mit denen die APT-Datenbank durchsucht und Informationen über die Pakete ausgegeben werden. Zusätzlich existieren einige Programme, die die Suche graphisch aufbereiten.

7.4.11.1 Paketsuche im Terminal Mit dem einfachen Befehl **`apt search <Suchmuster>`** erhält man die Liste aller Pakete, die das Suchmuster enthalten. Die Suche mit *search* erlaubt die Verwendung von regex-Begriffen.

Wird z. B. nach *“gman”* gesucht, erhält man dieses Ergebnis:

```
user1@pc1:~$ apt search ^gman
Sortierung... Fertig
Volltextsuche... Fertig
gman/unstable,now 0.9.3-5.3 amd64  [installiert]
  small man(1) front-end for X

gmanedit/unstable 0.4.2-7 amd64
  GTK+/GNOME-Editor für Handbuchseiten
```

Hier bedeutet das “^”, dass “gman” am Zeilenanfang stehen muss. Ohne dieses Zeichen findet das Muster beispielsweise auch khan*gman* und lo*gman*ager.

Möchte man mehr Informationen über die aktuellen Versionen eines Pakets, dann benutzt man:

```
user1@pc1:~$ apt show gman
Package: gman
Version: 0.9.3-5.3
Priority: optional
Section: doc
Maintainer: Josip Rodin <joy-packages@debian.org>
Installed-Size: 106 kB
Provides: man-browser
Depends: libc6 (>= 2.14), libgcc1 (>= 1:3.0), libglib2.0-0 ↙
   (>= 2.12.0),
 libgtk2.0-0 (>= 2.8.0), libstdc++6 (>= 5), man-db, xterm | x↙
    -terminal-emulator
Suggests: gv, man2html, httpd, sensible-browser, evince
Tag: implemented-in::c, interface::graphical, interface::web,↙
    interface::x11,
 role::program, uitoolkit::gtk, use::browsing, use::viewing, ↙
    web::cgi,
 works-with-format::html, works-with-format::man, works-with↙
    ::text,
```

```
 x11::application
Download-Size: 34,3 kB
APT-Manual-Installed: yes
APT-Sources: http://ftp.de.debian.org/debian unstable/main ⇙
   amd64 Packages
Description: small man(1) front-end for X
 Gman is a simple front-end for the manual page system. The ⇙
    most basic job
 of gman is to build a database for all the man pages and ⇙
    display them
 (or part of them) on the screen. When user decides to read a⇙
     man page,
 gman will launch an external viewer to display the manual ⇙
    page. More than
 one external viewer windows can be launched at the same time⇙
    .
 ...
```

Alle installierbaren Versionen des Pakets (abhängig von der sources.list) können folgendermaßen aufgelistet werden:

```
user1@pc1:~$ apt list gman
Auflistung... Fertig
gman/unstable,now 0.9.3-5.3 amd64  [installiert]
```

Der Befehl **aptitude** (im Terminal) öffnet das gleichnamige Programm in einer ncurses-Umgebung. Es wird mit der Tastatur oder Maus bedient und bietet diverse Funktionen, die über die obere Menüleiste erreichbar sind. Die Nutzung von APT oder Aptitude ist Geschmackssache, allerdings ist Aptitude für das Tempo von Debian Unstable oft “zu schlau”.

7.4.11.2 Graphische Paketsuche

Das Programm **packagesearch** eignet sich hervorragend um nach geeigneten Programmen zu suchen. Meist wird packagesearch nicht automatisch installiert; deshalb:

```
apt update
apt install packagesearch
```

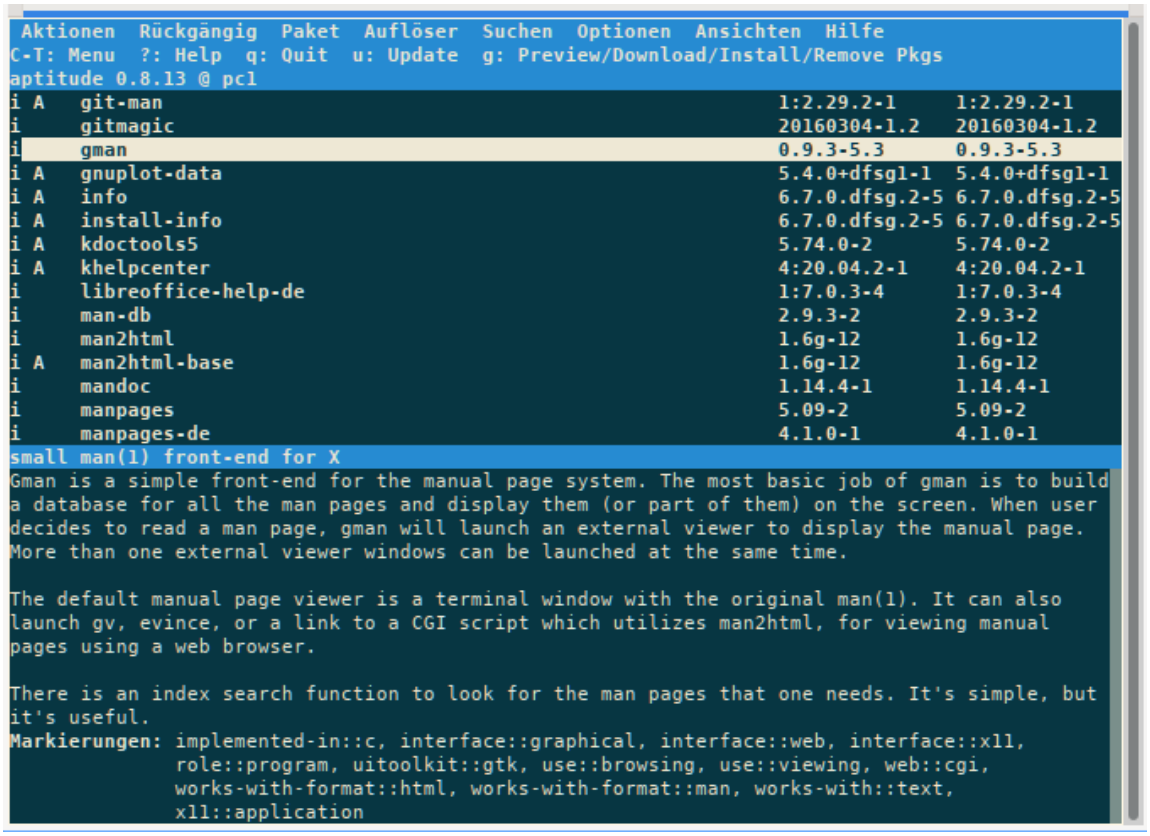


Abbildung 52: aptitude

Nach dem ersten Start von packagesearch muss in *Packagesearch* > *Preferences* “apt” gewählt werden und gelegentlich erscheint ein Infofenster, das das Fehlen von deborphan bemängelt. Die Informationen von deborphan bitte mit größter Vorsicht verwenden.

Packagesearch soll nicht zur Installation von Dateien/Paketen benutzt werden, sondern nur als eine graphische Suchmaschine. Das Upgraden und die Neuinstallation von Dateien ohne vorheriges Beenden von X kann Probleme verursachen (siehe oben).

Folgende Suchkriterien stehen zur Auswahl:

- pattern (allgemeine Suchanfrage)
- tags (Suche basierend auf debtags)
- files (Dateinamen)
- installed status (Installationsstatus)
- orphaned packages (verwaiste Pakete)

Zusätzlich werden viele Informationen zu den Debian-Paketen angeboten, so auch welche Dateien in einem Paket geschnürt sind. Weitere ausführliche Informationen zur Verwendung von packagesearch findet man unter *Help* > *Contents*. Derzeit ist die Benutzerführung von packagesearch ausschließlich Englisch.

Eine vollständige Beschreibung des APT-Systems findet man in Debians APT-HOWTO

Zuletzt bearbeitet: 2021-11-29

7.5 Lokaler APT-Mirror

Apt-Cacher, ein Proxy-Server für Debian-Pakete

Apt-Cacher ist ein Proxy-Server, der mehreren lokalen Computern den Zugang zu einem Debian-Pakete-Cache ermöglicht.
Die zur Installation von einem Computer bei dem Cache angeforderten Pakete müssen nur einmal von Debian Spiegelservern geladen werden, gleichgültig wie viele Geräte diese Pakete benötigen. Dies spart Netzwerkbandbreite, erhöht die Geschwindigkeit für die Benutzer und reduziert die Last an den Spiegelservern.

Nutzern, die mehrere PC ihr Eigen nennen und sparsam mit Bandbreite und Downloadvolumen umgehen und gleichzeitig die Geschwindigkeit bei Systemaktualisierungen erhöhen wollen, bietet apt-cacher die ideale Lösung um all diese Ziele zu erreichen.

Apt-Cacher ist kein universeller Proxy-Server. Wer dies trotzdem versucht, wird einige unliebsame Überraschungen im Netzwerk erleben.

Voraussetzungen

- Ein PC, auf dem der lokale APT-Proxy-Server eingerichtet wird.
- 6 GB freier Speicherplatz für den Cache auf dem Server.
- LAN-Verbindung zu den anderen Geräten.

Apt-Cacher Setup

Das Setup für Apt-Cacher erfolgt in zwei Schritten.
Als Erstes wird Apt-Cacher auf dem als APT-Proxy-Server ausgewählten PC installiert und anschließend konfiguriert man alle Client-PCs so, dass sie den APT-Proxy-Server benutzen.

7.5.1 Server installieren

Nach einem `apt update` werden die notwendigen Pakete mit folgendem Befehl installiert:

```
# apt install apt-cacher
[...]
Die folgenden NEUEN Pakete werden installiert:
  apt-cacher ed libberkeleydb-perl libcompress-raw-bzip2-perl
```

```
  libcompress-raw-lzma-perl libcompress-raw-zlib-perl   ↙
     libfilesys-df-perl
  libio-compress-lzma-perl libio-compress-perl libio-↙
     interactive-perl
  libio-interface-perl libipc-shareable-perl libnetaddr-ip-↙
     perl libsocket6-perl
  libsys-syscall-perl libwww-curl-perl
0 aktualisiert, 16 neu installiert, 0 zu entfernen und 0 ↙
   nicht aktualisiert.
Es müssen 992 kB an Archiven heruntergeladen werden.
Nach dieser Operation werden 3.205 kB Plattenplatz zusätzlich↙
    benutzt.
Möchten Sie fortfahren? [J]
```

Während der Installation von apt-cacher erfolgt automatisch die grundlegende Konfiguration.

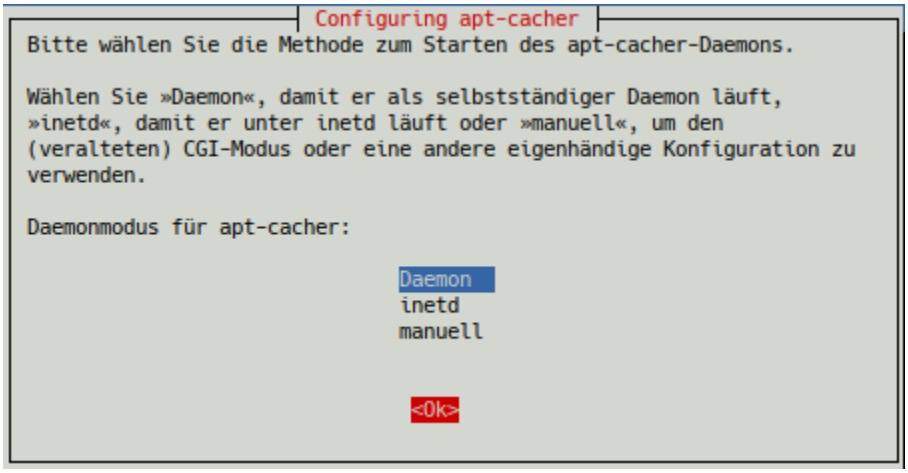


Abbildung 53: Konfiguration von apt-cacher

Der empfohlene Deamonmodus *“deamon”* wird beibehalten und bestätigt.

Der Cache, in dem in Zukunft alle heruntergeladenen Pakete abgelegt werden, befindet sich in
`/var/cache/apt-cacher/`
und die Konfigurationsdateien in
`/etc/apt-cacher/` .

Server Konfiguration

Wir wechseln in das Verzeichnis `/etc/apt-cacher/` und bearbeiten die Datei `apt-cacher.conf`.

```
# cd /etc/apt-cacher
/etc/apt-cacher# mcedit apt-cacher.conf
```

Jetzt suchen wir etwa bei Zeile 160 die Direktive `allowed_hosts`. Das Kommentarzeichen "#" am Anfang der Zeile wird entfernt, damit die Clients den APT-Proxy-Server kontaktieren dürfen.
Aus Sicherheitsgründen ersetzen wir das Platzhalterzeichen "*", das allen den Zugriff erlaubt, durch die IP-Adressen der Clients.

```
#allowed_hosts = *
```

beispielsweise ändern in

```
allowed_hosts = '192.168.3.10-20'
```

Die IP-Adressen sind natürlich an die eigenen Gegebenheiten anzupassen. Erläuterungen zur Syntax befinden sich in der Datei unmittelbar vor der Direktive.
Wird im eigenen Netzwerk ein DHCP-Server betrieben, so ist es notwendig dem APT-Proxy-Server eine feste IP zuzuordnen, z.B. "192.168.3.5" .

Wichtig für uns sind aus der Datei `apt-cacher.conf` der *"user"* und die *"group"* mit der der Daemon läuft und der *"port"* auf den der Daemon lauscht:

```
group = www-data
user = www-data
daemon_port = 3142
```

Das sind die voreingestellten Werte, die wir nicht verändern. Nach dem Speichern der Datei beenden wir mcedit.

Soll ein anderes Cache-Verzeichnis als `/var/cache/apt-cacher/` Verwendung finden, müssen die Eigentümer- und Dateirechte geprüft und angepasst werden (chmod 644 für die Dateien).

Um sicher zu gehen, dass der APT-Proxy-Server bei jedem Boot des Servers automatisch startet, setzen wir folgenden Befehl ab:

```
# systemctl enable --now apt-cacher.service
```

Der APT-Proxy-Server wird jetzt auch neu gestartet und damit die geänderte Konfiguration eingelesen.
Wir überprüfen ob er aktiv ist und auf Port 3142 lauscht.

```
# ss -tl | grep 3142
LISTEN 0      4096         0.0.0.0:3142        0.0.0.0:
```

Bei dieser Ausgabe ist alles in Ordnung.

Import vorhandener .deb Pakete

Apt-Cacher verfügt jetzt über ein Importscript, das auf dem PC bereits vorhandene Debian Archive importiert. Es erspart den nochmaligen Download der Pakete. Dem Aufruf geben wir das vorhandene Archivverzeichnis mit:

```
# /usr/share/apt-cacher/apt-cacher-import.pl  /var/cache/apt/↙
   archives/
```

Mit `-h` aufgerufen erhalten wir Benutzungshinweise und eine Auflistung aller Optionen.

7.5.2 Client Konfiguration

Die Clients, die auf den APT-Proxy-Server zugreifen, bedürfen nur geringfügiger Konfigurationen.

Zuerst legen wir die Datei `30proxy` im Verzeichnis `/etc/apt/apt.conf.d/` an, die *apt* anweist den Server zu benutzen. Wir verwenden hier die oben genannte IP des Servers. Bitte die IP an die eigenen Gegebenheiten anpassen.

```
# echo "Acquire::http { Proxy "http://192.168.3.5:3142"; };" ↙
   > /etc/apt/apt.conf.d/30proxy
```

Als nächstes ändern wir die Adressen der Downloadmirror im Verzeichnis `/etc↙/apt/sources.list.d/` von “https” auf “http” innerhalb der Dateien `debian.↙list`, `extra.list` und `fixes.list`.
Die Verwendung von “https” ist zwar möglich, aber zum Einen mit einigem Konfigurationsaufwand verbunden und zum Anderen zur Zeit nicht notwendig, da alle Downloadmirror “http” noch akzeptieren.

Ein anschließendes

```
# apt update
```

sollte ohne Fehlermeldungen durchlaufen.
Der erste Aufruf von **`apt full-upgrade`** auf einem Client lädt alle neuen Pakete in den Cache des APT-Proxy-Servers. Somit dauert dieser Vorgang genauso lang wie es zuvor üblich war. Die weiteren Zugriffe der Clienten bedienen sich des Cache und laufen dann wesentlich schneller ab, ohne erneut Bandbreite zu benötigen

Seite zuletzt aktualisiert 2021-11-29

7.6 Kernel Upgrade

Siduction stellt folgende Kernel bereit:

- **linux-image-siduction-amd64 + linux-headers-siduction-amd64** - Linux Kernel für 64-bit PCs mit AMD64 oder Intel 64 CPU.
- 32 bit Kernel stellen wir nicht mehr zur Verfügung, hier kann der Debian Kernel, oder alternativ der Liquorix-Kernel (https://liquorix.net/) verwendet werden.

Die Kernel von siduction befinden sich im siduction-Repository als .deb und werden bei einer Systemaktualisierung automatisch berücksichtigt, sofern die Metapakete für Image und Headers installiert sind.

7.6.1 Kernel-Aktualisierung ohne Systemaktualisierung

1. Aktualisierung der Paketdatenbank:

   ```
   apt update
   ```

2. Installation des aktuellen Kernels:

   ```
   apt install linux-image-siduction-amd64 linux-headers-↙
      siduction-amd64
   ```

3. Neustart des Computers, um den neuen Kernel zu laden.

   Falls sich mit dem neuen Kernel Probleme zeigen, kann man nach einem Neustart einen älteren Kernel wählen.

7.6.2 Module

Der Kernel bringt in der Regel alle benötigten Kernel-Module mit. Für 3rd Party Module wird in siduction dkms empfohlen. Hierzu ist es notwendig, das Paket `build↙-essential` zu installieren. Da 3rd Party Module oftmals unfreie Module sind, ist sicherzustellen, dass contrib und non-free in den Sourcen aktiviert ist.

7.6.3 Entfernen alter Kernel

Nach erfolgreicher Installation eines neuen Kernels können alte Kernel entfernt werden. Es ist jedoch empfohlen, alte Kernel einige Tage zu behalten. Falls mit

dem neuen Kernel Probleme auftauchen, kann in einen der alten Kernel gebootet werden, welche im Grub-Startbildschirm gelistet sind.

Zur Entfernung alter Kernel ist das Skript `kernel-remover` installiert:

```
kernel-remover
```

Seite zuletzt aktualisiert 2021-11-29

7.7 Systemd, der System- und Dienste-Manager

Anmerkung:
Die folgende, allgemeine Einführung zu systemd wurde überwiegend der ins deutsche übersetzten Manpage entnommen. Der Dank geht an Helge Kreutzmann.

systemd ist ein System- und Diensteverwalter, der beim Systemstart als erster Prozess (als PID 1) ausgeführt wird und somit als **Init-System** agiert, das System hochfährt und auf Anwendungsebene Dienste verwaltet.
Entwickelt wird es federführend von den Red Hat Entwicklern Lennart Poettering und Kay Sievers.

In Debian wurde die Einführung des systemd als Standard-Init-System lange, kontrovers und emotional diskutiert bis im Februar 2014 der Technische Ausschuss für systemd stimmte.

Seit der Veröffentlichung von 2013.2 “December” benutzt siduction bereits systemd als Standard-Init-System.

7.7.1 Konzeption des systemd

Systemd stellt ein Abhängigkeitssystem zwischen verschiedenen Einheiten namens *“Units”* in 11 verschiedenen Typen (siehe unten) bereit. Units kapseln verschiedene Objekte, die für den Systemstart und -betrieb relevant sind.
Units können *“aktiv”* oder *“inaktiv”*, sowie im Prozess der *“Aktivierung”* oder *“Deaktivierung”*, d.h. zwischen den zwei erstgenannten Zuständen sein. Ein besonderer Zustand *“fehlgeschlagen”* ist auch verfügbar, der sehr ähnlich zu inaktiv ist. Falls dieser Zustand erreicht wird, wird die Ursache für spätere Einsichtnahme protokolliert. Siehe die Handbuchseite Sytemd-Journal.
Mit systemd können viele Prozesse parallel gesteuert werden, da die Unit-Dateien mögliche Abhängigkeiten deklarieren und systemd erforderliche Abhängigkeiten automatisch hinzugefügt.

Die von systemd verwalteten Units werden mittels Unit-Dateien konfiguriert.
Die Unit-Dateien sind in verschiedene Sektionen unterteilte, reine Textdateien im INI-Format. Dadurch ist ihr Inhalt ohne Kenntnis einer Scriptsprache leicht verständlich und editierbar. Alle Unit-Dateien müssen eine Sektion entsprechend des Unit Typs haben und können die generischen Sektionen [Unit] und [Install] enthalten.

Die Handbuchseite Systemd Unit-Datei erläutert den grundlegenden Aufbau der Unit-Dateien, sowie viele Optionen der generischen Sektionen [Unit] und [Install].

7.7.2 Unit Typen

Bevor wir uns den Unit-Typen zuwenden, ist es ratsam die Handbuchseite Systemd Unit-Datei zu lesen, um die Wirkungsweise der generischen Sektionen und ihrer Optionen zu verstehen.
Die folgenden Unit-Typen sind verfügbar, und sofern verlinkt, führt der Link zu einer ausführlicheren Beschreibung in unserem Handbuch:

1. **Dienste-Units** (systemd.service), die Daemons und die Prozesse, aus denen sie bestehen, starten und steuern.
2. **Socket-Units** (systemd.socket), die lokale IPC- oder Netzwerk-Sockets im System kapseln, nützlich für Socket-basierte Aktivierung.
3. **Target-Units** (systemd.target) sind für die Gruppierung von Units nützlich. Sie stellen während des Systemstarts auch als Runlevel bekannte Synchronisationspunkte zur Verfügung.
4. **Geräte-Units** (systemd.device) legen Kernel-Geräte (alle Block- und Netzwerkgeräte) in systemd offen und können zur Implementierung Geräte-basierter Aktivierung verwandt werden.
5. **Mount-Units** (systemd.mount) steuern Einhängepunkte im Dateisystem.
6. **Automount-Units** (systemd.automount) stellen Fähigkeiten zum Selbsteinhängen bereit, für bedarfsgesteuertes Einhängen von Dateisystemen sowie parallelisiertem Systemstart.
7. **Zeitgeber-Units** (systemd.timer) sind für das Auslösen der Aktivierung von anderen Units basierend auf Zeitgebern nützlich.
8. **Auslagerungs-Units** (systemd.swap) sind ähnlich zu Einhänge-Units und kapseln Speicherauslagerungs-Partitionen oder -Dateien des Betriebssystems.
9. **Pfad-Units** (systemd.path) können zur Aktivierung andere Dienste, wenn sich Dateisystemobjekte ändern oder verändert werden, verwandt werden.

10. **Slice-Units** (systemd.slice) können zur Gruppierung von Units, die Systemprozesse (wie Dienste- und Bereichs-Units) in einem hierarchischen Baum aus Ressourcenverwaltungsgründen verwalten, verwandt werden.
11. **Scope-Units** (systemd.scope) sind ähnlich zu Dienste-Units, verwalten aber fremde Prozesse, statt sie auch zu starten.

7.7.3 Systemd im Dateisystem

Die Unit-Dateien, die durch den Paketverwalter der Distribution installiert wurden, befinden sich im Verzeichnis `/lib/systemd/system/`. Selbst erstellte Unit-Dateien legen wir im Verzeichnis `/usr/local/lib/systemd/system/` ab. (Ggf. ist das Verzeichnis zuvor mit dem Befehl **`mkdir -p /usr/local/lib/systemd/system/`** anzulegen.)
Die Steuerung des Status (enabled, disabled) einer Unit erfolgt über Symlink im Verzeichnis `/etc/systemd/system/`.
Das Verzeichnis `/run/systemd/system/` beinhaltet zur Laufzeit erstellte Unit-Dateien.

7.7.4 Weitere Funktionen von systemd

Systemd bietet noch weitere Funktionen. Eine davon ist logind als Ersatz für das nicht mehr weiter gepflegte *ConsoleKit* . Damit steuert systemd Sitzungen und Energiemanagement. Nicht zuletzt bietet systemd eine Menge an weiteren Möglichkeiten wie beispielsweise das Aufspannen eines Containers (ähnlich einer Chroot) mittels systemd-nspawn und viele weitere. Ein Blick in die Linkliste auf Freedesktop ermöglicht weitere Entdeckungen, unter anderem auch die ausführliche Dokumentation von Hauptentwickler Lennart Poettering zu systemd.

7.7.5 Handhabung von Diensten

Einer der Jobs von systemd ist es Dienste zu starten, zu stoppen oder sonst wie zu steuern. Dazu dient der Befehl `systemctl`.

- systemctl –all - listet alle Units, aktive und inaktive.
- systemctl -t [NAME] - listet nur Units des bezeichneten Typ.
- systemctl list-units - listet alle aktiven Units.
- systemctl start [NAME…] - startet eine oder mehrere Units.

- systemctl stop [NAME…] - stoppt eine oder mehrere Units.
- systemctl restart [NAME] - stoppt eine Unit und startet sie sofort wieder. Wird z.B. verwendet um die geänderte Konfiguration eines Dienstes neu einzulesen.
- systemctl status [Name] - zeigt den derzeitigen Status einer Unit.
- systemctl is-enabled [Name] - zeigt nur den Wert “enabled” oder “disabled” des Status einer Unit.

Die beiden folgenden Befehle integrieren bzw. entfernen die Unit anhand der Konfiguration ihrer Unit-Datei. Dabei werden Abhängigkeiten zu anderen Units beachtet und ggf. Standardabhängigkeiten hinzugefügt, damit systemd die Dienste und Prozesse fehlerfrei ausführen kann.

- systemctl enable [NAME] - gliedert eine Unit in systemd ein.
- systemctl disable [NAME] - entfernt eine Unit aus systemd.

Oft ist es nötig, *“systemctl start”* und *“systemctl enable”* für eine Unit durchzuführen, um sie sowohl sofort als auch nach einem Reboot verfügbar zu machen. Beide Optionen vereint der Befehl:

- systemctl enable –now [NAME]

Nachfolgend zwei Befehle deren Funktion unsere Handbuchseite Systemd-Target beschreibt.

- systemctl reboot – Führt einen Reboot aus
- systemctl poweroff - Fährt das System herunter und schaltet den Strom, sofern technisch möglich, aus.

Beispiel
Anhand von Bluetooth demonstrieren wir die Funktionalität des systemd.
Zuerst die Statusabfrage im Kurzformat.

```
# systemctl is-enabled bluetooth.service
enabled
```

Nun Suchen wir nach den Unit-Dateien, dabei kombinieren wir *“systemctl”* mit *“grep”*:

```
# systemctl list-unit-files | grep blue
bluetooth.service          enabled         enabled
dbus-org.bluez.service     alias           -
```

```
bluetooth.target           static          -
```

Anschließend deaktivieren wir die Unit *“bluetooth.service”*.

```
# systemctl disable bluetooth.service
  Synchronizing state of bluetooth.service with SysV service ↙
     script with /lib/systemd/systemd-sysv-install.
  Executing: /lib/systemd/systemd-sysv-install disable ↙
     bluetooth
  Removed /etc/systemd/system/dbus-org.bluez.service.
  Removed /etc/systemd/system/bluetooth.target.wants/↙
     bluetooth.service.
```

In der Ausgabe ist gut zu erkennen, dass die Link (nicht die Unit-Datei selbst) entfernt wurden. Damit startet der *“bluetooth.service”* beim Booten des PC/Laptop nicht mehr automatisch. Zur Kontrolle fragen wir den Status nach einem Reboot ab.

```
# systemctl is-enabled bluetooth.service
disabled
```

Um eine Unit nur zeitweise zu deaktivieren, verwenden wir den Befehl

```
# systemctl stop <unit>
```

Damit bleibt die Konfiguration in systemd erhalten. Mit dem entsprechenden “start”-Befehl aktivieren wir die Unit wieder.

7.7.6 Quellen systemd

Deutsche Manpage ‘systemd’
Deutsche Manpage ‘systemd.unit’
Deutsche Manpage ‘systemd.syntax’

Seite zuletzt aktualisiert 2021-11-29

7.8 systemd unit-Datei

Die grundlegenden und einführenden Informationen zu Systemd enthält die Handbuchseite Systemd-Start
In der vorliegenden Handbuchseite erklären wir den Aufbau der **Unit-Dateien** und die generischen Sektionen *[Unit]* und *[Install]*.

Die Unit-Datei ist eine reine Textdatei im INI-Format. Sie enthält Konfigurationsanweisungen von der Art “Schlüssel=Wert” in verschiedenen Sektionen. Leere Zeilen und solche, die mit “#” oder “;” beginnen, werden ignoriert. Alle Unit-Dateien müssen eine Sektion entsprechend des Unit-Typ enthalten. Die generischen Sektionen [Unit] am Beginn und [Install] am Ende der Datei sind optional, wobei die Sektion [Unit] dringend empfohlen wird.

7.8.1 Ladepfad der Unit-Dateien

Die Ausgabe zeigt die Reihenfolge der Verzeichnisse, aus denen die Unit-Dateien geladen werden.

```
# systemd-analyze unit-paths
/etc/systemd/system.control
/run/systemd/system.control
/run/systemd/transient
/run/systemd/generator.early
/etc/systemd/system
/etc/systemd/system.attached
/run/systemd/system
/run/systemd/system.attached
/run/systemd/generator
/usr/local/lib/systemd/system
/lib/systemd/system
/usr/lib/systemd/system
/run/systemd/generator.late
```

Unit-Dateien, die in früher aufgeführten Verzeichnissen gefunden werden, setzen Dateien mit dem gleichen Namen in Verzeichnissen, die weiter unten in der Liste aufgeführt sind, außer Kraft. So hat eine Datei in `/etc/systemd/system` Vorrang vor der gleichnamigen in `/lib/systemd/system`.

Nur ein Teil der zuvor aufgeführten Verzeichnisse existiert per default in siduction. Die Verzeichnisse

- `/lib/systemd/system/`
  beinhalten System-Units, die durch den Paketverwalter der Distribution installiert wurden und ggf. vom Administrator erstellte Unit-Dateien.

- `/etc/systemd/system/`
  beinhalten Symlinks auf Unit-Dateien in `/lib/systemd/system/` für aktivierte Units und ggf. vom Administrator erstellte Unit-Dateien.

- `/usr/local/lib/systemd/system/`
  dieses Verzeichnis muss erstellt werden und ist für vom Administrator erstellte Unit-Dateien vorgesehen.

- `/run/systemd/`
  beinhalten Laufzeit-Units und dynamische Konfiguration für flüchtige Units. Für den Administrator hat dieses Verzeichnis ausschließlich informellen Wert.

Wir empfehlen eigene Unit-Dateien in `/usr/local/lib/systemd/system/` abzulegen.

7.8.2 Aktivierung der Unit-Datei

Um systemd die Konfiguration einer Unit zugänglich zu machen, muss die Unit-Datei aktiviert werden. Dies geschieht mit dem Aufruf:

```
# systemctl daemon-reload
# systemctl enable --now <UNIT_DATEI>
```

Der erste Befehl lädt die komplette Daemon-Konfiguration neu, der zweite startet die Unit sofort (Option `--now`) und gliedert sie in systemd ein, sodass sie bei jedem Neustart des PC ausgeführt wird.
Der Befehl

```
# systemctl disable <UNIT_DATEI>
```

bewirkt, dass sie nicht mehr bei jedem Neustart des PC ausgeführt wird. Sie kann aber weiterhin manuell mit dem Befehl **`systemctl start <UNIT_DATEI>`** gestartet und mit **`systemctl stop <UNIT_DATEI>`** gestoppt werden.
Falls eine Unit-Datei leer ist (d.h. die Größe 0 hat) oder ein Symlink auf `/dev/null` ist, wird ihre Konfiguration nicht geladen und sie erscheint mit einem Ladezustand “masked” und kann nicht aktiviert werden. Dies ist eine wirksame Methode um eine Unit komplett zu deaktivieren und es auch unmöglich zu machen, sie manuell zu starten.

7.8.3 Sektionen der Unit-Datei

Die Unit-Datei besteht in der Regel aus der Sektionen [Unit], der Typ-spezifischen Sektion und der Sektion [Install]. Die Typ-spezifische Sektion fließt als Suffix in den Dateinamen ein. So besitzt zum Beispiel eine Unit-Datei, die einen Zeitgeber konfiguriert, immer die Endung “.timer” und muss [Timer] als Typ-spezifische Sektion enthalten.

7.8.3.1 Sektion Unit Diese Sektion enthält allgemeine Informationen über die Unit, definiert Abhängigkeiten zu anderen Units, wertet Bedingungen aus und sorgt für die Einreihung in den Bootprozess.

1. Allgemeine Optionen

    a. `Description=`
    Identifiziert die Unit durch einen menschenlesbaren Namen, der von systemd als Bezeichnung für die Unit verwandt wird und somit im systemjournal erscheint (“Starting *description*…”) und dort als Suchmuster verwandt werden kann.

    b. `Documentation=`
    Ein Verweis auf eine Datei oder Webseite, die Dokumentation für diese Unit oder ihre Konfiguration referenzieren. Z. B.: “Documentation=man:cupsd(8)” oder “Documentation=http://www.cups.org/doc/man-cupsd.html”.

2. Bindungsabhängigkeiten zu anderen Units

    a. `Wants=`
    Hier aufgeführte Units werden mit der konfigurierten Unit gestartet.

b. `Requires=`
   Ähnlich zu Wants=, erklärt aber eine stärkere Bindung an die aufgeführten Units.
   Wenn diese Unit aktiviert wird, werden die aufgeführten Units ebenfalls aktiviert.
   Schlägt die Aktivierung einer der anderen Units fehl **und** die Ordnungsabhängigkeit `After=` ist auf die fehlgeschlagene Unit gesetzt, dann wird diese Unit nicht gestartet.
   Falls eine der anderen Units inaktiv wird, bleibt diese Unit aktiv, nur wenn eine der anderen Units gestoppt wird, wird diese Unit auch gestoppt.

c. `Requisite=`
   Ähnlich zu Requires=. Der Start dieser Unit wird sofort fehlschlagen, wenn die hier aufgeführten Units noch nicht gestartet wurden. Requisite= sollte mit der Ordnungsabhängigkeit `After=` kombiniert werden, um sicherzustellen, dass diese Unit nicht vor der anderen Unit gestartet wird.

d. `BindsTo=`
   BindsTo= ist der stärkste Abhängigkeitstyp: Es bewirkt zusätzlich zu den Eigenschaften von Requires=, dass die gebundene Unit im aktiven Status sein muss, damit diese Unit auch aktiv sein kann.
   Beim Stoppen oder inaktivem Zustand der gebundenen Unit wird diese Unit immer gestoppt.
   Um zu verhindern, dass der Start dieser Unit fehlschlägt, wenn die gebundene Unit nicht, oder noch nicht in einem aktiven Zustand ist, sollte BindsTo= am besten mit der Ordnungsabhängigkeit `After=` kombiniert werden.

e. `PartOf=`
   Ähnlich zu Requires=, aber begrenzt auf das Stoppen und Neustarten von Units.
   Wenn Systemd die hier aufgeführten Units stoppt oder neu startet, wird die Aktion zu dieser Unit weitergeleitet.
   Das ist eine Einwege-Abhängigkeit. Änderungen an dieser Unit betreffen nicht die aufgeführten Units.

f. `Conflicts=`
   Deklariert negative Anforderungs-Abhängigkeiten. Die Angabe einer durch Leerzeichen getrennten Liste ist möglich.
   Conflicts= bewirkt, dass die aufgeführte Unit gestoppt wird, wenn diese Unit startet und umgekehrt.
   Da Conflicts= keine Ordnungs-Abhängigkeit beinhaltet, muss eine Abhängigkeit `After=` oder `Before=` erklärt werden, um sicherzustellen, dass die in Konflikt stehende Unit gestoppt wird, bevor die andere Unit gestartet wird.

3. Ordnungsabhängigkeiten zu anderen Units

   a. `Before=`
      Diese Einstellung konfiguriert Ordnungsabhängigkeiten zwischen Units. Before= stellt sicher, dass die aufgeführte Unit erst mit dem Starten beginnt, nachdem der Start der konfigurierten Unit abgeschlossen ist.
      Die Angabe einer durch Leerzeichen getrennten Liste ist möglich.

   b. `After=`
      Diese Einstellung stellt das Gegenteil von Before= sicher. Die aufgeführte Unit muss vollständig gestartet sein, bevor die konfigurierte Unit gestartet wird.

   c. `OnFailure=`
      Units, die aktiviert werden, wenn diese Unit den Zustand “failed” einnimmt.

4. Bedingungen
   Unit-Dateien können auch eine Reihe von Bedingungen enthalten.
   Bevor die Unit gestartet wird, wird Systemd nachweisen, dass die festgelegten Bedingungen wahr sind. Falls nicht, wird das Starten der Unit (fast ohne Ausgabe) übersprungen.
   Fehlschlagende Bedingungen führen nicht dazu, dass die Unit in den Zustand »failed« überführt wird.
   Falls mehrere Bedingungen festgelegt sind, wird die Unit ausgeführt, falls alle von ihnen zutreffen.
   In diesem Abschnitt führen wir nur Bedingungen auf, die uns für selbst erstellte Units hilfreich erscheinen, denn viele Bedingungen dienen dazu, um Units zu überspringen, die auf dem lokalen System nicht zutreffen.

Der Befehl **`systemd-analyze verify <UNIT_DATEI>`** kann zum Testen von Bedingungen verwandt werden.

a. `ConditionVirtualization=`
   Prüft, ob das System in einer virtualisierten Umgebung ausgeführt wird und testet optional, ob es eine bestimmte Implementierung ist.

b. `ConditionACPower=`
   Prüft, ob das System zum Zeitpunkt der Aktivierung der Unit am Netz hängt oder ausschließlich über Akku läuft.

c. `ConditionPathExists=`
   Prüft auf die Existenz einer Datei. Mit einem Ausrufezeichen (“!”) vor dem Pfad wird der Test negiert.

d. `ConditionPathExistsGlob=`
   Wie zuvor, nur dass ein Suchmuster angegeben wird. Mit einem Ausrufezeichen (“!”) vor dem Pfad wird der Test negiert.

e. `ConditionPathIsDirectory=`
   Prüft auf die Existenz eines Verzeichnisses. Mit einem Ausrufezeichen (“!”) vor dem Pfad wird der Test negiert.

f. `ConditionPathIsSymbolicLink=`
   Überprüft ob ein bestimmter Pfad existiert und ein symbolischer Link ist. Mit einem Ausrufezeichen (“!”) vor dem Pfad wird der Test negiert.

g. `ConditionPathIsMountPoint=`
   Überprüft ob ein bestimmter Pfad existiert und ein Einhängepunkt ist. Mit einem Ausrufezeichen (“!”) vor dem Pfad wird der Test negiert.

h. `ConditionPathIsReadWrite=`
   Überprüft ob das zugrundeliegende Dateisystem les- und schreibbar ist. Mit einem Ausrufezeichen (“!”) vor dem Pfad wird der Test negiert.

i. `ConditionDirectoryNotEmpty=`
   Überprüft ob ein bestimmter Pfad existiert und ein nicht leeres Verzeichnis ist. Mit einem Ausrufezeichen (“!”) vor dem Pfad wird der Test negiert.

j. `ConditionFileNotEmpty=`
   Überprüft ob ein bestimmter Pfad existiert und sich auf eine normale

Datei mit einer von Null verschiedenen Größe bezieht. Mit einem Ausrufezeichen (“!”) vor dem Pfad wird der Test negiert.

k. `ConditionFileIsExecutable=`
Überprüft ob ein bestimmter Pfad existiert und sich auf eine normale, als ausführbar gekennzeichnete Datei bezieht. Mit einem Ausrufezeichen (“!”) vor dem Pfad wird der Test negiert.

Die vollständige Dokumentation zu allen Optionen der Sektion [Unit] bitte in der Deutschen Manpage, systemd.unit nachlesen.

7.8.3.2 Typ-spezifische Sektion Diese Sektion enthält die speziellen Optionen der elf möglichen Typen. Ausführliche Beschreibungen enthalten die verlinkten Handbuchseiten, oder ersatzweise die jeweilige deutsche Manpage.

- [Service] konfiguriert einen Dienst
- [Socket] konfiguriert ein Socket
- [Device] konfiguriert ein Gerät
- [Mount] konfiguriert einen Einhängepunkt
- [Automount] konfiguriert einen Selbsteinhängepunkt
- [Swap] konfiguriert eine Auslagerungsdatei oder -partition
- [Target] konfiguriert ein Startziel
- [Path] konfiguriert einen überwachten Dateipfad
- [Timer] konfiguriert einen von systemd gesteuerten und überwachten Zeitgeber
- [Slice] konfiguriert eine Ressourcenverwaltungs-Slice
- [Scope] konfiguriert eine Gruppe von extern erstellten Prozessen.

7.8.3.3 Sektion Install Unit-Dateien können diese Sektion enthalten.
Die Optionen der [Install]-Sektion werden von den Befehlen **`systemctl enable <UNIT_DATEI>`** und **`systemctl disable <UNIT_DATEI>`** während der Installation einer Unit verwandt.

Unit-Dateien ohne [Install]-Sektion lassen sich manuell mit dem Befehl **`systemctl start <UNIT_DATEI>`**, oder von einer anderen Unit-Datei starten.

Beschreibung der Optionen:

- `Alias=`
  Eine Liste von zusätzlichen Namen, unter der diese Unit installiert werden soll. Die hier aufgeführten Namen müssen die gleiche Endung wie die Unit-Datei haben.

- `WantedBy=`
  Diese Option kann mehrfach verwendet werden oder eine durch Leerzeichen getrennte Liste enthalten.
  Im `.wants/`-Verzeichnis jeder der aufgeführten Units wird bei der Installation ein symbolischer Link erstellt. Dadurch wird eine Abhängigkeit vom Typ `Wants=` von der aufgeführten Unit zu der aktuellen Unit hinzugefügt. Das Hauptergebnis besteht darin, dass die aktuelle Unit gestartet wird, wenn die aufgeführte Unit gestartet wird.
  Verhält sich wie die Option Wants= in der Sektion [Unit].

  Beispiel:
  WantedBy=graphical.target

  Das teilt systemd mit, die Unit beim Starten von graphical.target (früher “init 5”) hereinzuziehen.

- `RequiredBy=`
  Diese Option kann mehrfach verwendet werden oder eine durch Leerzeichen getrennte Liste enthalten.
  Im `.requires/`-Verzeichnis jeder der aufgeführten Units wird bei der Installation ein symbolischer Link erstellt. Dadurch wird eine Abhängigkeit vom Typ `Requires=` von der aufgeführten Unit zu der aktuellen Unit hinzugefügt. Das Hauptergebnis besteht darin, dass die aktuelle Unit gestartet wird, wenn die aufgeführte Unit gestartet wird.
  Verhält sich wie die Option Requires= in der Sektion [Unit].

- `Also=`
  Zusätzliche Units, die installiert/deinstalliert werden sollen, wenn diese Unit installiert/deinstalliert wird.

- `DefaultInstance=`
  Diese Option zeigt nur bei Vorlagen-Unit-Dateien Wirkung.
  Deklariert, welche Instanz der Unit freigegeben werden soll. Die angegebene Zeichenkette muss zur Identifizierung einer Instanz geeignet sein.

Hinweis: Um die Konfiguration einer Unit-Datei zu prüfen, eignet sich der Befehl **`systemd-analyze verify <UNIT_DATEI>`**.

7.8.4 Beispiel cupsd

Der *“cupsd”*, Auftragsplaner (Scheduler) für das Common UNIX Printing System, wird von systemd mit seinen drei Unit Dateien `cups.socket`, `cups.service` und `cups.path` gesteuert und eignet sich gut, um die Abhängigkeiten zu verdeutlichen. Hier die drei Dateien.

```
Datei /lib/systemd/system/cups.service:

[Unit]
Description=CUPS Scheduler
Documentation=man:cupsd(8)
After=network.target sssd.service ypbind.service nslcd.⤶
   service
Requires=cups.socket
    After=cups.socket (nicht in der Datei, da implizit ⤶
       vorhanden.)
    After=cups.path   (nicht in der Datei, da implizit ⤶
       vorhanden.)

[Service]
ExecStart=/usr/sbin/cupsd -l
Type=notify
Restart=on-failure

[Install]
Also=cups.socket cups.path
WantedBy=printer.target
```

```
Datei /lib/systemd/system/cups.path:
```

```
[Unit]
Description=CUPS Scheduler
PartOf=cups.service
    Before=cups.service (nicht in der Datei, da implizit ↙
       vorhanden.)

[Path]
PathExists=/var/cache/cups/org.cups.cupsd

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

```
Datei /lib/systemd/system/cups.socket:

[Unit]
Description=CUPS Scheduler
PartOf=cups.service
    Before=cups.service (nicht in der Datei, da implizit ↙
       vorhanden.)

[Socket]
ListenStream=/run/cups/cups.sock

[Install]
WantedBy=sockets.target
```

Die Sektion [Unit]

enthält für alle drei Dateien die gleiche Beschreibung. Die Dateien `cups.path` und `cups.socket` enthalten zusätzlich die Bindungsabhängigkeit `PartOf=cups.service`, was bedeutet, dass diese zwei Units abhängig von cups.service gestoppt oder neu gestartet werden.

Die socket-Unit ebenso wie die path-Unit schließen die Ordnungsabhängigkeit `Before=` zu ihrer namensgleichen service-Unit ein. Deshalb ist es nicht notwendig in der cups.service-Unit die Ordnungs-Abhängigkeiten After=cups.socket und After=cups.path einzutragen. (Siehe unten die Ausgabe von *"systemd-analyze dump"* mit dem Vermerk *"destination-implicit"*.) Beide Abhängigkeiten gemeinsam bewirken, dass unabhängig davon, welche Unit zuerst startet, immer alle drei Units

starten und die cups.service-Unit erst, nachdem der Start der cups.path-Unit und der cups.socket-Unit erfolgreich abgeschlossen wurde.

Die vollständige Konfiguration der Units erhalten wir mit dem Befehl **`systemd-analyze dump`**, der eine sehr, sehr lange Liste (> 32000 Zeilen) des systemd Serverstatus ausgibt.

```
# systemd-analyze dump
[...]
-> Unit cups.service:
    Description: CUPS Scheduler.service
    [...]
    WantedBy: printer.target (destination-file)
    ConsistsOf: cups.socket (destination-file)
    ConsistsOf: cups.path (destination-file)
    Before: printer.target (destination-default)
    After: cups.socket (destination-implicit)
    After: cups.path (destination-implicit)
[...]
-> Unit printer.target:
    Description: Printer
    [...]
    Wants: cups.service (origin-file)
    After: cups.service (origin-default)
[...]
```

Die Sektion [Install]

der cups.service-Unit enthält mit der Option `Also=cups.socket cups.path` die Anweisung, diese beiden Units auch zu installieren und alle drei Units haben unterschiedliche `WantedBy=` Optionen:

- cups.socket: WantedBy=sockets.target
- cups.path: WantedBy=multi-user.target
- cups.service: WantedBy=printer.target

Um zu verstehen, warum unterschiedliche Werte für die Option *"WantedBy="* Verwendung finden, benötigen wir zusätzliche Informationen, die wir mit den Befehlen **`systemd-analyze dot`** und **`systemd-analyze plot`** erhalten.

```
$ systemd-analyze dot --to-pattern='*.target' --from-pattern⬋
   =\
    '*.target' | dot -Tsvg > targets.svg

$ systemd-analyze plot > bootup.svg
```

Der erste liefert uns ein Flussdiagramm mit den Abhängigkeiten der verschiedenen Targets zueinander und der zweite eine graphisch aufbereitete Auflistung des Bootprozesses mit den Zeitpunkten wann ein Prozess gestartet wurde, welche Zeit er beanspruchte und seinen Aktivitätszustand.

Der `targets.svg` und der `bootup.svg` entnehmen wir, dass

1. *sysinit.target*
   aktiviert wird und
2. *basic.target*
   erst startet, wenn sysinit.target erreicht wurde.
   1. *sockets.target*
      von basic.target angefordert wird,
      1. *cups.socket*
         und alle weiteren .socket-Units von sockets.target hereingeholt werden.
   2. *paths.target*
      von basic.target angefordert wird,
      1. *cups.path*
         und alle weiteren .path-Units von paths.target hereingeholt werden.
3. *network.target*
   erst startet, wenn basic.target erreicht wurde.
4. *cups.service*
   erst startet, wenn network.target erreicht wurde.

5. *multi-user.target*
   erst startet, wenn network.target erreicht wurde.

6. *multi-user.target*
   erst dann erreicht wird, wenn cups.service erfolgreich gestartet wurde.
   (Genau genommen liegt es daran, dass der cups-browsed.service, der vom cups.service abhängt, erfolgreich gestartet sein muss.)

7. *printer.target*
   wird erst aktiv, wenn systemd dynamisch Geräte-Units für die Drucker generiert.
   Dazu müssen die Drucker angeschlossen und eingeschaltet sein.

Weiter oben stellten wir fest, dass der Start einer cups.xxx-Unit ausreicht, um alle drei Units hereinzuholen. Betrachten wir noch einmal die "*WantedBy=*"-Optionen in der [Install]-Sektion, so haben wir die cups.socket-Unit, die über das sockets.target bereits während des basic.target hereingeholt wird, die cups.path-Unit, die während des multi-user.target hereingeholt wird und den cups.service, der vom printer.target hereingeholt wird.
Während des gesamten Bootprozesses werden die drei cups.xxx-Units wiederholt bei systemd zur Aktivierung angefordert. Das härtet den cupsd gegen unvorhergesehene Fehler, spielt für systemd aber keine Rolle, denn es ist unerheblich wie oft ein Service angefordert wird, wenn er sich in der Warteschlange befindet.
Zusätzlich fordert immer dann das printer.target den cups.service an, wenn ein Drucker neu von systemd erkannt wird.

7.8.5 Werkzeuge

Systemd beinhaltet einige nützliche Werkzeuge für die Analyse, Prüfung und Bearbeitung der Unit-Dateien.
Bitte auch die Manpages systemd-analyze und systemctl zu Rate ziehen.

- edit

```
# systemctl edit <UNIT_DATEI>
# systemctl edit --full <UNIT_DATEI>
# systemctl edit --full --force <UNIT_DATEI>
```

  "systemctl edit" öffnet die ausgewählte Unit-Datei im konfigurierten Editor.

- **`systemctl edit <UNIT_DATEI>`**
  erstellt unterhalb `/etc/systemd/system/` ein neues Verzeichnis mit dem Namen `<UNIT_DATEI>.d` und darin die Datei `override.conf`, die ausschließlich die Änderungen gegenüber der ursprünglichen Unit-Datei enthält. Dies gilt für alle Unit-Dateien in den Verzeichnissen, die in der Hierarchie der Ladepfade inklusive `/etc/systemd/system/` abwärts eingetragen sind.
- **`systemctl edit --full <UNIT_DATEI>`**
  erstellt eine neue, namensgleiche Datei im Verzeichnis `/etc/systemd/system/`. Dies gilt für alle Unit-Dateien in den Verzeichnissen, die in der Hierarchie der Ladepfade unterhalb `/etc/systemd/system/` eingetragen sind. Dateien, die sich bereits im Verzeichnis `/etc/systemd/system/` befinden, werden überschrieben.
- **`systemctl edit --full --force <UNIT_DATEI>`**
  erstellt eine neue Datei im Verzeichnis `/etc/systemd/system/`. Ohne die Option `--full` würde nur eine Datei `override.conf` im neuen Verzeichnis `/etc/systemd/system/<UNIT_DATEI>.d` generiert, der die zugehörige Unit-Datei fehlt.

Wird der Editor beendet, so führt systemd automatisch den Befehl **`systemctl daemon-reload`** aus.

- revert

```
# systemctl revert <UNIT_DATEI>
```

macht die mit `systemctl edit` und `systemctl edit --full` vorgenommenen Änderungen an Unit-Dateien rückgängig. Dies gilt nicht für geänderte Unit-Dateien die sich bereits im Verzeichnis `/etc/systemd/system/` befanden.
Zusätzlich bewirkt der Befehl die Rücknahme der mit `systemctl mask` vorgenommenen Änderungen.

- daemon-reload

```
# systemctl daemon-reload
```

Lädt die Systemverwalterkonfiguration neu. Dies führt alle Generatoren neu aus, lädt alle Unit-Dateien neu und erstellt den gesamten Abhängigkeitsbaum neu.

- cat

```
$ systemctl cat <UNIT_DATEI>
```

Gibt entsprechend des Konsolebefehls `cat` den Inhalt der Unit-Datei und aller zugehörigen Änderungen aus.

- analyze verify

```
$ systemd-analyze verify <UNIT_DATEI>
```

überprüft die Konfigurationseinstellungen einer Unit-Datei und gibt Hinweise aus. Dies ist ein sehr hilfreicher Befehl um die Konfiguration selbst erstellter oder geänderter Unit-Dateien zu prüfen.

- systemd-delta

```
$ systemd-delta
```

präsentiert in der Ausgabe Unit-Dateien und die vorgenommenen Änderungen an ihnen. Das Schlüsselwort am Anfang der Zeile definiert die Art der Änderung bzw. Konfiguration.
Hier ein Beispiel:

```
$ systemd-delta --no-pager
[MASKED]     /etc/sysctl.d/50-coredump.conf → /usr/lib/↙
   sysctl.d/50-coredump.conf
[OVERRIDDEN] /etc/tmpfiles.d/screen-cleanup.conf → /usr/↙
   lib/tmpfiles.d/screen-cleanup.conf
[MASKED]     /etc/systemd/system/NetworkManager-wait-↙
   online.service → /lib/systemd/system/NetworkManager-↙
   wait-online.service
[EQUIVALENT] /etc/systemd/system/tmp.mount → /lib/systemd↙
   /system/tmp.mount
[EXTENDED]   /lib/systemd/system/rc-local.service → /lib/↙
   systemd/system/rc-local.service.d/debian.conf
```

```
[EXTENDED]   /lib/systemd/system/systemd-localed.service ⬋
   → /lib/systemd/system/systemd-localed.service.d/locale⬋
   -gen.conf

6 overridden configuration files found.
```

- analyze dump

```
$ systemd-analyze dump > systemd_dump.txt
```

erstellt die Textdatei *“systemd_dump.txt”* mit der vollständigen Konfiguration alle Units des systemd. Die sehr lange Textdatei gibt Aufschluss über alle Konfigurationseinstellungen aller systemd-Units und lässt sich mit einem Texteditor und unter Verwendung von RegEx-Pattern gut durchsuchen.

- analyze plot

```
$ systemd-analyze plot > bootup.svg
```

erstellt die Datei *“bootup.svg”* mit der zeitlichen Abfolge des Bootprozesses. Es ist eine graphisch aufbereitete Auflistung des Bootprozesses mit den Start- und Endzeitpunkten aller Units, welche Zeit sie beanspruchten und ihren Aktivitätszuständen.

- analyze dot

```
$ systemd-analyze dot --to-pattern='*.target' --from-⬋
   pattern=\
  '*.target' | dot -Tsvg > targets.svg
  Color legend: black     = Requires
                dark blue = Requisite
                dark grey = Wants
                red       = Conflicts
                green     = After
```

erstellt das Flussdiagramm *“targets.svg”*, dass die Abhängigkeiten der im Bootprozess verwendeten Targets darstellt. Die Beziehungen der target-Units werden zur besseren Übersicht farblich dargestellt.

Die hier genannten Hilfsmittel stellen nur einen Teil der mit systemd ausgelieferten Werkzeuge dar. Bitte entnehme den man-Pages die vollständige Dokumentation.

7.8.6 Quellen systemd-unit-Datei

Deutsche Manpage, systemd.unit
Deutsche Manpage, systemd.syntax
Deutsche Manpage, systemd.device
Deutsche Manpage, systemd.scope
Deutsche Manpage, systemd.slice
Deutsche Manpage, systemd.socket
Deutsche Manpage, systemd.swap
Deutsche Manpage, systemd-analyze
Deutsche Manpage, systemctl

Dank an Helge Kreuzmann für die deutschen Übersetzungen.

Seite zuletzt aktualisiert 2021-11-29

7.9 systemd-service

Die grundlegenden und einführenden Informationen zu Systemd enthält die Handbuchseite Systemd-Start Die alle Unit-Dateien betreffenden Sektionen *[Unit]* und *[Install]* behandelt unsere Handbuchseite Systemd Unit-Datei
In der vorliegenden Handbuchseite erklären wir die Funktion der Unit **systemd.service**. Die Unit-Datei mit der Namensendung ".service" ist der am häufigsten anzutreffende Unit-Typ in systemd.

Die Service-Unit-Datei muss eine Sektion [Service] enthalten, die Informationen über den Dienst und den Prozess, den er überwacht, konfiguriert.

7.9.1 service-Unit anlegen

Selbst erstellte Unit-Dateien legen wir vorzugsweise im Verzeichnis `/usr/local/lib/systemd/system/` ab. (Ggf. ist das Verzeichnis mit dem Befehl **`mkdir -p /usr/local/lib/systemd/system/`** anzulegen.) Das hat den Vorteil, dass sie Vorrang gegenüber den System-Units, die durch den Paketverwalter der Distribution installiert wurden, erhalten und gleichzeitig Steuerungslinks sowie Änderungsdateien, die mit **`systemctl edit <UNIT_DATEI>`** erzeugt wurden, im seinerseits vorrangigen Verzeichnis `/etc/systemd/system/` abgelegt werden. Siehe: Hierarchie der Ladepfade.

7.9.2 Sektion Service

Für diese Sektion sind über dreißig Optionen verfügbar, von denen wir hier besonders häufig verwendete beschreiben.

```
Type=             PIDFile=
RemainAfterExit=  GuessMainPID=
ExecStart=        Restart=
ExecStartPre=     RestartSec=
ExecStartPost=    SuccessExitStatus=
ExecCondition=    RestartPreventExitStatus=
ExecReload=       RestartForceExitStatus=
ExecStop=         NonBlocking=
ExecStopPost=     NotifyAccess=
TimeoutStopSec=   RootDirectoryStartOnly=
TimeoutStartSec=  FileDescriptorStoreMax=
```

```
TimeoutAbortSec=  USBFunctionDescriptors=
TimeoutSec=       USBFunctionStrings=
RuntimeMaxSec=    Sockets=
WatchdogSec=      BusName=
                  OOMPolicy=
```

- `Type=`
  Definiert den Prozess-Starttyp und ist damit eine der wichtigsten Optionen. Die möglichen Werte sind: simple, exec, forking, oneshot, dbus, notify oder idle.
  Der Standard simple wird verwendet, falls `ExecStart=` festgelegt ist, aber weder `Type=` noch `BusName=` gesetzt sind.
  - `simple`
    Eine Unit vom Typ simple betrachtet systemd als erfolgreich gestartet, sobald der mit `ExecStart=` festgelegte Hauptprozess mittels *fork* gestartet wurde. Anschließend beginnt systemd sofort mit dem Starten von nachfolgenden Units, unabhängig davon, ob der Hauptprozess erfolgreich aufgerufen werden kann.
  - `exec`
    Ähnelt simple, jedoch wartet systemd mit dem Starten von nachfolgenden Units bis der Hauptprozess erfolgreich beendet wurde. Das ist auch der Zeitpunkt, an dem die Unit den Zustand “active” erreicht.
  - `forking`
    Hier betrachtet systemd den Dienst als gestartet, sobald der mit `ExecStart=` festgelegte Prozess sich in den Hintergrund verzweigt und das übergeordnete System sich beendet. Dieser Typ findet oft bei klassischen Daemons Anwendung. Hier sollte auch die Option `PIDFile=` angeben werden, damit das System den Hauptprozess weiter verfolgen kann.
  - `oneshot`
    Ähnelt exec. Die Option Type=oneshot kommt oft bei Skripten oder Befehlen zum Einsatz, die einen einzelnen Job erledigen und sich dann beenden. Allerdings erreicht der Dienst niemals den Zustand “active”, sondern geht sofort, nachdem sich der Hauptprozess beendet hat, vom

Zustand “activating” zu “deactivating” oder “dead” über. Deshalb ist es häufig sinnvoll diese Option mit `RemainAfterExit=yes` zu verwenden, um den Zustand “active” zu erreichen.

  - `dbus`
    Verhält sich ähnlich zu simple, systemd startet nachfolgende Units, nachdem der D-Bus-Busname erlangt wurde. Units mit dieser Option, erhalten implizit eine Abhängigkeit auf die Unit dbus.socket.

  - `notify`
    Der Type=notify entspricht weitestgehend dem Type simple, mit dem Unterschied, dass der Daemon ein Signal an systemd sendet, wenn er bereitsteht.

  - `idle`
    Das Verhalten von idle ist sehr ähnlich zu simple; allerdings verzögert systemd die tatsächliche Ausführung des Dienstes, bis alle aktiven Aufträge erledigt sind. Dieser Typ ist nicht als allgemeines Werkzeug zum Sortieren von Units nützlich, denn er unterliegt einer Zeitüberschreitung von 5 s, nach der der Dienst auf jeden Fall ausgeführt wird.

- `RemainAfterExit=`
  Erwartet einen logischen Wert (Standard: no), der festlegt, ob der Dienst, selbst wenn sich alle seine Prozesse beendet haben, als aktiv betrachtet werden sollte. Siehe Type=oneshot.

- `GuessMainPID=`
  Erwartet einen logischen Wert (Standard: yes). Systemd verwendet diese Option ausschließlich, wenn `Type=forking` gesetzt und `PIDFile=` nicht gesetzt ist, und versucht dann die Haupt-PID eines Dienstes zu raten, falls es sie nicht zuverlässig bestimmen kann. Für andere Typen oder mit gesetzter Option `PIDFile=` ist die Haupt-PID immer bekannt.

- `PIDFile=`
  Akzeptiert einen Pfad zur PID-Datei des Dienstes. Für Dienste vom `Type=forking` wird die Verwendung dieser Option empfohlen.

- `BusName=`
  Hier ist der D-Bus-Busname, unter dem dieser Dienst erreichbar ist, anzugeben. Die Option ist für Dienste vom `Type=dbus` verpflichtend.

- `ExecStart=`
  Enthält Befehle mit ihren Argumenten, die ausgeführt werden, wenn diese Unit gestartet wird. Es muss genau ein Befehl angegeben werden, außer die Option `Type=oneshot` ist gesetzt, dann kann ExecStart= mehrfach verwendet werden. Der Wert von ExecStart= muss den in der deutsche Manpage systemd.service detailliert beschriebenen Regeln entsprechen.

- `ExecStop=`
  Kann mehrfach verwendet werden und enthält Befehle, die dem Stoppen eines mittels ExecStart= gestarteten Dienstes, dienen. Die Syntax ist identisch zu ExecStart=.

- `ExecStartPre=`, `ExecStartPost=`, `ExecStopPost=`
  Zusätzliche Befehle, die vor bzw. nach dem Befehl in `ExecStart=` oder `ExecStop=` gestartet werden. Auch hier ist die Syntax identisch zu ExecStart=. Es sind mehrere Befehlszeilen erlaubt und die Befehle werden seriell einer nach dem anderen ausgeführt. Falls einer dieser Befehle (dem nicht "-" vorangestellt ist) fehlschlägt, wird die Unit sofort als fehlgeschlagen betrachtet.

- `RestartSec=`
  Bestimmt die vor dem Neustart eines Dienstes zu schlafende Zeit. Eine einheitenfreie Ganzzahl definiert Sekunden, eine Angabe von "3min 4s" ist auch möglich.
  Die Art der Zeitwertdefinition gilt für alle zeitgesteuerten Optionen.

- `TimeoutStartSec=`, `TimeoutStopSec=`, `TimeoutSec=`
  Bestimmt die Zeit, die auf das Starten bzw. Stoppen gewartet werden soll. TimeoutSec= vereint die beiden zuvor genannten Optionen.
  TimeoutStopSec= konfiguriert zusätzlich die Zeit, die, soweit vorhanden, für jeden `ExecStop=`-Befehl gewartet werden soll.

- `Restart=`
  Konfiguriert, ob der Dienst neu gestartet werden soll, wenn der Dienstepro-zess sich beendet, getötet oder eine Zeitüberschreitung erreicht wird. Wenn der Tod des Prozesses das Ergebnis einer Systemd-Aktion ist, wird der Dienst nicht neu gestartet.
  Die erlaubten Werte sind: no, always, on-success, on-failure, on-abnormal, on-abort oder on-watchdog.

Folgende Tabelle zeigt den Effekt der Restart= Einstellung auf die Exit-Gründe.

► Restart= ► ▼ Exit-Grund ▼	always	on success	on failure	on abnormal	on abort	on watchdog
Sauberer Exit	X	X				
Unsauberer Exit	X		X			
Unsauberes Signal	X		X	X	X	
Zeitüberschreitung	X		X	X		
Watchdog	X		X	X		X

Die bei Bedarf gesetzten Optionen `RestartPreventExitStatus=` und `RestartForceExitStatus=` ändern dieses Verhalten.

Beispiele

Einige selbst erstellte Service-Units finden sich auf unseren Handbuchseiten

service-Unit für systemd Timer
service-Unit für systemd Path
und mit der bevorzugten Suchmaschine im Internet.
LinuxCommunity, Systemd-Units selbst erstellen

7.9.3 Quellen systemd-service

Deutsche Manpage, systemd.service
LinuxCommunity, Systemd-Units selbst erstellen

Seite zuletzt aktualisiert 2021-11-29

7.10 systemd-mount

Die grundlegenden und einführenden Informationen zu Systemd enthält die Handbuchseite Systemd-Start Die alle Unit-Dateien betreffenden Sektionen *[Unit]* und *[Install]* behandelt unsere Handbuchseite Systemd Unit-Datei
In der vorliegenden Handbuchseite erklären wir die Funktion der systemd-Units **mount** und **automount**. Mit ihnen verwaltet systemd Einhängepunkte für Laufwerke und deren Partitionen, die sowohl lokal als auch über das Netzwerk erreichbar sein können.

Die **mount**-Unit ist eine Konfigurationsdatei, die für systemd Informationen über einen Einhängepunkt bereitstellt.
Die **automount**-Unit überwacht das Dateisystem und aktiviert die gleichnamige mount-Unit, wenn das darin bezeichnete Dateisystem verfügbar ist.

Für unmittelbar im PC verbaute Laufwerke und deren Partitionen verwenden wir nur die mount-Unit. Sie wird aktiviert (enabled) und gestartet um die Laufwerke bei jedem Boot einzuhängen.
Bei Netzwerk-Dateisystemen bietet die mount-Unit den Vorteil, Abhängigkeiten deklarieren zu können, damit die Unit erst aktiv wird, wenn das Netzwerk bereit steht. Auch hier benutzen wir nur die mount-Unit und aktivieren und starten sie, um das Netzwerk-Dateisystemen bei jedem Boot einzuhängen. Die mount-Unit unterstützt alle Arten von Netzwerk-Dateisystemen (NFS, SMB, FTP, WEBDAV, SFTP, SSH).

Entfernbare Geräte, wie USB-Sticks, und Netzwerk-Dateisysteme, die nicht permanent erreichbar sind, müssen immer an eine automount-Unit gekoppelt werden. In diesem Fall darf die mount-Unit nicht aktiviert werden und sollte auch keine [Install]-Sektion enthalten.

mount- und automount-Units müssen nach dem Einhängepunkt, den sie steuern, benannt sein. Beispiel: Der Einhängepunkt “/home/musteruser” muss in einer Unit-Datei “home-musteruser.mount”, bzw. “home-musteruser.automount”, konfiguriert werden.

Die in der `/etc/fstab` deklarierten Geräte und ihre Einhängepunkte übersetzt systemd in der frühen Bootphase mit Hilfe des systemd-fstab-generators in native mount-Units.

7.10.1 Inhalt der mount-Unit

Die mount-Unit verfügt über die folgenden Optionen in der zwingend erforderlichen [Mount]-Sektion:

- `What=` (Pflicht)
  Enthält den absoluten Pfad des eingehängten Geräts, also z.B. Festplatten-Partitionen wie `/dev/sda8` oder eine Netzwerkfreigabe wie NFSv4 oder Samba.
- `Where=` (Pflicht)
  Hier wird der Einhängepunkt (mount point) festgelegt, d.h. der Ordner, in den die Partition, das Netzlaufwerk oder Gerät eingehängt werden soll. Falls dieser nicht existiert, wird er beim Einhängen erzeugt.
- `Type=` (optional)
  Hier wird der Typ des Dateisystems angegeben, gemäß dem mount-Parameter -t.
- `Options=` (optional)
  Enthält alle verwendeten Optionen in einer Komma getrennten Liste, gemäß dem mount-Parameter -o.
- `LazyUmount=` (Standard: off)
  Wenn der Wert auf true gesetzt wird, wird das Dateisystem wieder ausgehängt, sobald es nicht mehr benötigt wird.
- `SloppyOptions=` (Standard: off)
  Falls true, erfolgt eine entspannte Auswertung der in `Options=` festgelegten Optionen und unbekannte Einhängeoptionen werden toleriert. Dies entspricht dem mount-Parameter -s.
- `ReadWriteOnly=` (Standard: off)
  Falls false, wird bei dem Dateisystem oder Gerät, das read-write eingehängt werden soll, das Einhängen aber scheitert, versucht es read-only einzuhängen. Falls true, endet der Prozess sofort mit einem Fehler, wenn die Einhängung read-write scheitert. Dies entspricht dem mount-Parameter -w.
- `ForceUnmount=` (Standard: off)
  Falls true, wird das Aushängen erzwungen wenn z. B. ein NFS-Dateisystem nicht erreichbar ist. Dies entspricht dem mount-Parameter -f.

- `DirectoryMode=` (Standard: 0755)
  Die, falls notwendig, automatisch erzeugten Verzeichnisse von Einhängepunkten, erhalten den deklarierten Dateisystemzugriffsmodus. Akzeptiert einen Zugriffsmodus in oktaler Notation.

- `TimeoutSec=` (Vorgabewert aus der Option `DefaultTimeoutStartSec=` in systemd-system.conf)
  Konfiguriert die Zeit, die auf das Beenden des Einhängebefehls gewartet wird. Falls ein Befehl sich nicht innerhalb der konfigurierten Zeit beendet, wird die Einhängung als fehlgeschlagen betrachtet und wieder heruntergefahren. Akzeptiert einen einheitenfreien Wert in Sekunden oder einen Zeitdauerwert wie “5min 20s”. Durch Übergabe von “0” wird die Zeitüberschreitungslogik deaktiviert.

7.10.2 Inhalt der automount-Unit

Die automount-Unit verfügt über die folgenden Optionen in der zwingend erforderlichen [Automount]-Sektion:

- `Where=` (Pflicht)
  Hier wird der Einhängepunkt (mount point) festgelegt, d.h. der Ordner, in den die Partition, das Netzlaufwerk oder Gerät eingehängt werden soll. Falls dieser nicht existiert, wird er beim Einhängen erzeugt.

- `DirectoryMode=` (Standard: 0755)
  Die, falls notwendig, automatisch erzeugten Verzeichnisse von Einhängepunkten erhalten den deklarierten Dateisystemzugriffsmodus. Akzeptiert einen Zugriffsmodus in oktaler Notation.

- `TimeoutIdleSec=` (Standard: 0)
  Bestimmt die Zeit der Inaktivität, nach der systemd versucht das Dateisystem auszuhängen. Akzeptiert einen einheitenfreien Wert in Sekunden oder einen Zeitdauerwert wie “5min 20s”. Der Wert “0” deaktiviert die Option.

7.10.3 Beispiele

Systemd liest den Einhängepunkt aus dem Namen der mount- und automount-Units. Deshalb müssen sie nach dem Einhängepunkt, den sie steuern, benannt sein.

Dabei ist zu beachten, keine Bindestriche “-” in den Dateinamen zu verwenden, denn sie deklarieren ein neues Unterverzeichnis im Verzeichnisbaum. Einige Beispiele:

- unzulässig: /data/home-backup
- zulässig: /data/home_backup
- zulässig: /data/home\x2dbackup

Um einen fehlerfreien Dateinamen für die mount- und automount-Unit zu erhalten, verwenden wir im Terminal den Befehl `systemd-escape`.

```
$ systemd-escape -p --suffix=mount "/data/home-backup"
  data/home\x2dbackup.mount
```

Festplatten-Partition

Eine Partition soll nach jedem Systemstart unter `/disks/TEST` erreichbar sein. Wir erstellen mit einem Texteditor die Datei `disks-TEST.mount` im Verzeichnis `/usr/local/lib/systemd/system/`. (Ggf. ist das Verzeichnis zuvor mit dem Befehl **`mkdir -p /usr/local/lib/systemd/system/`** anzulegen.)

```
[Unit]
Description=Mount /dev/sdb7 at /disks/TEST
After=blockdev@dev-disk-by\x2duuid-a7af4b19\x2df29d\x2d43bc\⸗
   x2d3b12\x2d87924fc3d8c7.target
Requires=local-fs.target
Wants=multi-user.target

[Mount]
Where=/disks/TEST
What=/dev/disk/by-uuid/a7af4b19-f29d-43bc-3b12-87924fc3d8c7
Type=ext4
Options=defaults,noatime

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

Anschließend aktivieren und starten wir die neue mount-Unit.

```
# systemctl enable --now disks-TEST.mount
```

NFS

Das “document-root”-Verzeichnis eines Apache Webservers im heimischen Netzwerk soll in das Home-Verzeichnis des Arbeitsplatz-Rechners mittels NFS eingehängt werden.
Wir erstellen mit einem Texteditor die Datei `home-<user>-www_data.mount` im Verzeichnis `/usr/local/lib/systemd/system/`.
“<user>” bitte mit dem eigenen Namen ersetzen.

```
[Unit]
Description=Mount server1/var/www/ using NFS
After=network-online.target
Wants=network-online.target

[Mount]
What=192.168.3.1:/
Where=/home/<user>/www_data
Type=nfs
Options=nfsvers=4,rw,users,soft
ForceUnmount=true
```

Diese Datei enthält keine [Install]-Sektion und wird auch nicht aktiviert. Die Steuerung übernimmt die nun folgende Datei “home-<user>-www_data.automount” im gleichen Verzeichnis.

```
[Unit]
Description=Automount server1/var/www/ using NFS
ConditionPathExists=/home/<user>/www_data
Requires=NetworkManager.service
After=network-online.target
Wants=network-online.target

[Automount]
Where=/home/<user>/www_data
TimeoutIdleSec=60

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

Anschließend:

```
# systemctl enable --now home-<user>-www_data.automount
```

Jetzt wird das “document-root”-Verzeichnis des Apache Webservers eingehangen, sobald wir in das Verzeichnis `/home/\<user\>/www_data` wechseln.
Die Statusabfrage bestätigt die Aktion.

```
# systemctl status home-<user>-www_data.mount --no-pager●
 home-<user>-www_data.mount - Mount server1/var/www/ using ⇙
    NFS
     Loaded: loaded (/usr/local/lib/systemd/system/home-<user⇙
        >-www_data.mount; disabled; vendor preset: enabled)
     Active: active (mounted) since Wed 2021-03-10 16:27:58 ⇙
        CET; 8min ago
TriggeredBy: ● home-<user>-www_data.automount
      Where: /home/<user>/www_data
       What: 192.168.3.1:/
      Tasks: 0 (limit: 4279)
     Memory: 120.0K
        CPU: 5ms
     CGroup: /system.slice/home-<user>-www_data.mount
[...]

# systemctl status home-<user>-www_data.automount --no-pager●
 home-<user>-www_data.automount - Automount server1/var/www/ ⇙
    usuing NFS
     Loaded: loaded (/usr/local/lib/systemd/system/home-<user⇙
        >-www_data.automount; enabled; vendor preset: enabled⇙
        )
     Active: active (running) since Wed 2021-03-10 16:27:58 ⇙
        CET; 8min ago
   Triggers: ● home-<user>-www_data.mount
      Where: /home/<user>/www_data
[...]
```

Der Journalauszug protokolliert anschaulich die Funktion von *“TimeoutIdleSec=60”* zum Aushängen des Dateisystems und das wieder Einhängen durch

den Start des Dateimanagers Thunar sowie einen Aufruf von `/home/\<user\>/www_data` im Terminal.

```
# journalctl -f -u home-<user>-www_data.*
[...]
Mär 10 17:56:14 pc1 systemd[1]: Mounted Mount server1/var/www/ using NFS.
Mär 10 17:57:34 pc1 systemd[1]: Unmounting Mount server1/var/www/ using NFS...
Mär 10 17:57:35 pc1 systemd[1]: home-<user>-www_data.mount: Succeeded.
Mär 10 17:57:35 pc1 systemd[1]: Unmounted Mount server1/var/www/ using NFS.
Mär 10 17:58:14 pc1 systemd[1]: home-<user>-www_data.automount: Got automount request for /home/<user>/www_data, triggered by 2500 (Thunar)
Mär 10 17:58:14 pc1 systemd[1]: Mounting Mount server1/var/www/ using NFS...
Mär 10 17:58:14 pc1 systemd[1]: Mounted Mount server1/var/www/ using NFS.
Mär 10 18:00:15 pc1 systemd[1]: Unmounting Mount server1/var/www/ using NFS...
Mär 10 18:00:15 pc1 systemd[1]: home-<user>-www_data.mount: Succeeded.
Mär 10 18:00:15 pc1 systemd[1]: Unmounted Mount server1/var/www/ using NFS.
Mär 10 18:00:30 pc1 systemd[1]: home-<user>-www_data.automount: Got automount request for /home/<user>/www_data, triggered by 6582 (bash)
Mär 10 18:00:30 pc1 systemd[1]: Mounting Mount server1/var/www/ using NFS...
Mär 10 18:00:30 pc1 systemd[1]: Mounted Mount server1/var/www/ using NFS.
Mär 10 18:01:51 pc1 systemd[1]: Unmounting Mount server1/var/www/ using NFS...
Mär 10 18:01:51 pc1 systemd[1]: home-<user>-www_data.mount: Succeeded.
Mär 10 18:01:51 pc1 systemd[1]: Unmounted Mount server1/var/www/ using NFS.
```

```
[...]
```

Weitere Beispiele

Im Internet finden sich mit Hilfe der favorisierten Suchmaschine vielerlei Beispiele für die Anwendung der mount- und automount-Unit. Das Kapitel “Quellen” enthält einige Webseiten mit einer ganzen Reihe weiterer Beispiele. Dringend empfohlen sind auch die man-Pages.

7.10.4 Quellen systemd-mount

Deutsche Manpage, systemd.mount
Deutsche Manpage, mount
Manjaro Forum, systemd.mount
Manjaro Forum, Use systemd to mount ANY device
Linuxnews, nfs per systemd
Debianforum, Netzlaufwerke einbinden
Ubuntuusers, Mount-Units

Seite zuletzt aktualisiert 2021-11-29

7.11 systemd-target - Ziel-Unit

Die grundlegenden und einführenden Informationen zu Systemd enthält die Handbuchseite Systemd-Start Die alle Unit-Dateien betreffenden Sektionen *[Unit]* und *[Install]* behandelt unsere Handbuchseite Systemd Unit-Datei.
Jetzt erklären wir die Funktion der Unit **systemd.target**, die den allgemein bekannten Runleveln ähneln, etwas ausführlicher.

Die verschiedenen Runlevel, in die gebootet oder gewechselt wird, beschreibt systemd als Ziel-Unit. Sie besitzen die Erweiterung ".target".

Die alten sysvinit-Befehle werden weiterhin unterstützt. (Hierzu ein Zitat aus `man↙ systemd`: "… wird aus Kompatibilitätsgründen und da es leichter zu tippen ist, bereitgestellt.")

Ziel-Unit	Beschreibung
emergency.target	Startet in eine Notfall-Shell auf der Hauptkonsole. Es ist die minimalste Version eines Systemstarts, um eine interaktive Shell zu erlangen. Mit dieser Unit kann der Bootvorgang Schritt für Schritt begleitet werden.
rescue.target	Startet das Basissystem (einschließlich Systemeinhängungen) und eine Notfall-Shell. Im Vergleich zu multi-user.target könnte dieses Ziel als single-user.target betrachtet werden.
multi-user.target	Mehrbenutzersystem mit funktionierendem Netzwerk, ohne Grafikserver X. Diese Unit wird verwendet, wenn man X stoppen bzw. nicht in X booten möchte. Auf dieser Unit wird in besonderen Fällen (wenn X selbst oder die Desktop-Umgebung aktualisiert werden) eine Systemaktualisierung (dist-upgrade) durchgeführt .
graphical.target	Die Unit für den Mehrbenutzermodus mit Netzwerkfähigkeit und einem laufenden X-Window-System.
default.target	Die Vorgabe-Unit, die Systemd beim Systemstart startet. In siduction ist dies ein Symlink auf graphical.target (außer bei der Variante noX).

Ein Blick in die Dokumentation `man SYSTEMD.SPECIAL(7)` ist obligatorisch um die Zusammenhänge der verschiedenen target-Unit zu verstehen.

7.11.1 Besonderheiten

Bei den Ziel-Units sind drei Besonderheiten zu beachten:

1. Die Verwendung auf der Kernel-Befehszeile beim Bootvorgang.
   Um im Bootmanager Grub in den Editiermodus zu gelangen, muss man beim Erscheinen der Bootauswahl die Taste **e** drücken. Anschließend hängt man an die Kernel-Befehszeile das gewünschte Ziel mit der folgenden Syntax: “systemd.unit=xxxxxxx.target” an. Die Tabelle listet die Kernel-Befehle und ihre noch gültigen numerischen Entsprechungen auf.

Ziel-Unit	Kernel-Befehl	Kernel-Befehl alt
emergency.target	systemd.unit=emergency.target	-
rescue.target	systemd.unit=rescue.target	1
multi-user.target	systemd.unit=multi-user.target	3
graphical.target	systemd.unit=graphical.target	5

   Die alten Runlevel 2 und 4 verweisen auf multi-user.target

2. Die Verwendung im Terminal während einer laufenden Sitzung. Vorausgesetzt man befindet sich in einer laufenden graphischen Sitzung, kann man mit der Tastenkombination **CTRL**+**ALT**+**F2** zum virtuellen Terminal tty2 wechseln. Hier meldet man sich als User **root** an. Die folgende Tabelle listet die Terminal-Befehle auf, wobei der Ausdruck *“isolate”* dafür sorgt, dass alle Dienste die die Ziel-Unit nicht anfordert, beendet werden.

Ziel-Unit	Terminal-Befehl	init-Befehl alt
emergency.target	systemctl isolate emergency.target	-
rescue.target	systemctl isolate rescue.target	init 1
multi-user.target	systemctl isolate multi-user.target	init 3
graphical.target	systemctl isolate graphical.target	init 5

3. Ziel-Units, die nicht direkt aufgerufen werden sollen.
   Eine ganze Reihe von Ziel-Units sind dazu da während des Bootvorgangs

oder des .target-Wechsels Zwischenschritte mit Abhängigkeiten zu gruppieren. Die folgende Liste zeigt drei häufig verwendete Kommandos die **nicht** mit der Syntax “isolate xxxxxxx.target” aufgerufen werden sollen.

Ziel	Terminal-Befehl	init-Befehl alt
halt	systemctl halt	-
poweroff	systemctl poweroff	init 0
reboot	systemctl reboot	init 6

“halt”, *“poweroff”* und *“reboot”* holen mehrere Units in der richtigen Reihenfolge herein, um das System geordnet zu beenden und ggf. einen Neustart auszuführen.

7.11.2 Quellen systemd-target

Manpage systemd.target, de

Seite zuletzt aktualisiert 2021-11-29

7.12 systemd-path

Die grundlegenden und einführenden Informationen zu Systemd enthält die Handbuchseite Systemd-Start Die alle Unit-Dateien betreffenden Sektionen *[Unit]* und *[Install]* behandelt unsere Handbuchseite Systemd Unit-Datei.
In der vorliegenden Handbuchseite erklären wir die Funktion der Unit **systemd.path**, mit der systemd Pfade überwacht und Pfad-basierte Aktionen auslöst.

Die path-Unit ermöglicht es, bei Änderungen an Dateien und Verzeichnissen (Pfaden) eine Aktion auszulösen.
Sobald ein Ereignis eintritt, kann Systemd einen Befehl oder ein Skript über eine Service-Unit ausführen. Die path-Unit ist nicht in der Lage Verzeichnisse rekursiv zu überwachen. Es können aber mehrere Verzeichnisse und Dateien angegeben werden.
Die Pfad-spezifischen Optionen werden in dem Abschnitt [Path] konfiguriert.

7.12.1 Benötigte Dateien

Die **systemd-path**-Unit benötigt für ihre Funktion mindestens zwei Dateien mit vorzugsweise dem gleichen Namen, aber unterschiedlicher Namenserweiterung im Verzeichnis `/usr/local/lib/systemd/system/`. (Ggf. ist das Verzeichnis zuvor mit dem Befehl **`mkdir -p /usr/local/lib/systemd/system/`** anzulegen.) Das sind die

- Path-Unit-Datei (<name>.path), welche die Überwachung und den Auslöser für die Service-Unit enthält
  und

- Service-Unit-Datei (<name>.service), welche die zu startende Aktion enthält. Für umfangreichere Aktionen erstellt man zusätzlich ein Skript in `/usr/local/bin/`, das von der Service-Unit ausgeführt wird.

7.12.2 path-Unit Optionen

Die path-Unit muss zwingend die Sektion [Path] enthalten, in der festgelegt wird wie und was zu überwachen ist.

Die speziellen Optionen sind:

- `PathExists=`
  prüft, ob der betreffende Pfad existiert. Wenn es zutrifft, wird die zugehörige Unit aktiviert.
- `PathExistsGlob=`
  Wie oben, unterstützt Datei-Glob-Ausdrücke (siehe dazu auch die Ausgabe von man glob.
- `PathChanged=`
  beobachtet eine Datei oder einen Pfad und aktiviert die zugehörige Unit, wenn Änderungen auftreten.
  Aktionsauslösende Änderungen sind:
  - Erstellen und Löschen von Dateien.
  - Atribute, Rechte, Eigentümer.
  - Schließen der zu beobachtenden Datei nach Schreibzugriff und Schließen irgendeiner Datei nach Schreibzugriff bei Beobachtung des Pfades.
- `PathModified=`
  wie zuvor, aber zusätzlich wird die zugehörige Unit bei einfachen Schreibzugriffen aktiviert, auch wenn die Datei nicht geschlossen wird.
- `DirectoryNotEmpty=`
  aktiviert die zugehörige Unit wenn das Verzeichnis nicht leer ist.
- `Unit=`
  die zu aktivierende, zugehörige Unit. Zu beachten ist auch, dass die path-Unit standardmäßig die service-Unit mit dem gleichen Name aktiviert. Nur bei Abweichungen hiervon ist die Option `Unit=` innerhalb der Sektion [Path] notwendig.
- `MakeDirectory=`
  das zu beobachtenden Verzeichnis wird vor der Beobachtung erstellt.
- `DirectoryMode=`
  legt bei Verwendung für das zuvor erstellte Verzeichnis den Zugriffsmodus in oktaler Notation fest. Standardmäßig 0755.

Ein Beispiel

Basierend auf der Konfiguration des Apache-Webservers, entsprechend unserer Handbuchseite LAMP - Apache, Benutzer und Rechte, wollen wir das Zusammenspiel der path-Unit mit anderen systemd-Unit verdeutlichen.

Die Abbildung *path-Unit-Funktion* stellt die Abhängigkeiten der systemd-Units unseres Beispiels dar.

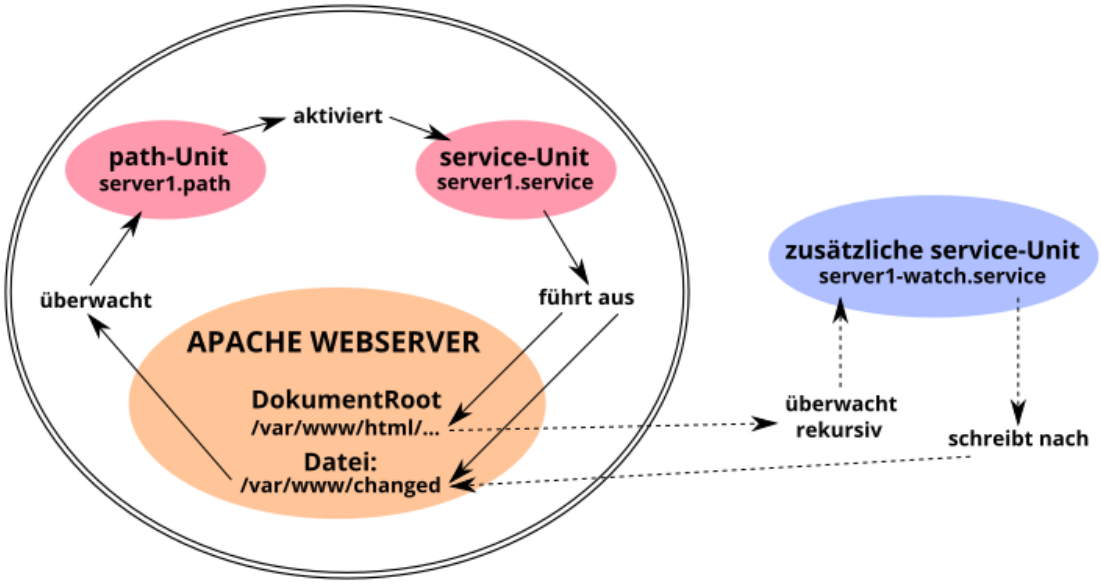


Abbildung 54: path-Unit Funktion

Der doppelt umrandete Teil in der Graphik verdeutlicht die Kernfunktion der path-Unit. Die server1.path-Unit überwacht die Datei `/var/www/changed` und aktiviert bei Änderungen die zugehörige server1.service-Unit. Diese wiederum führt dann die gewünschten Aktionen im Verzeichnis `/var/www/html/` aus und stellt die Datei `/var/www/changed` zurück.
Die außerhalb der Umrandung liegende server1-watch.service-Unit übernimmt die rekursive Überwachung von *DocumentRoot* des Apache-Webservers.

7.12.3 path-Unit anlegen

Wir legen die Datei `server1.path` im Verzeichnis `/usr/local/lib/systemd/system/`, die die Datei `/var/www/changed` auf Änderungen überwacht, mit folgendem Inhalt an:

```
[Unit]
Description=Monitoring "changed" file!
```

```
BindsTo=server1-watch.service
After=server1-watch.service

[Path]
PathModified=/var/www/changed

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

Erklärungen

Sektion [Unit]:

Die Option *"BindsTo="* stellt die stärkste verfügbare Bindung zweier systemd-Einheiten aneinander dar. Falls eine von ihnen während des Starts oder des Betriebs in einen Fehlerzustand übergeht, wird die andere auch unmittelbar beendet. Zusammen mit der Option *"After="* wird erreicht, dass die server1.path-Unit erst startet, nachdem die server1-watch.service-Unit ihren erfolgreichen Start an systemd zurückmeldet.

Sektion [Path]:

"PathModifid=" ist die richtige Wahl. Die Option reagiert auf Änderungen in der Datei `/var/www/changed`, selbst wenn die Datei nicht geschlossen wird.

Die Option *"PathModifid="* (oder andere, siehe oben) kann mehrfach angegeben werden.

7.12.4 service-Unit für path

Die server1.service-Unit wird von der server1.path-Unit aktiviert und kontrolliert und benötigt daher keine [Install] Sektion. Somit reichen die Beschreibung der Unit in der Sektion [Unit], und in der Sektion [Service] die auszuführenden Befehle, aus.

Wir legen die Datei `server1.service` im Verzeichnis `/usr/local/lib/systemd/system/` mit folgendem Inhalt an.

```
[Unit]
Description=Change permissions in server1 folder

[Service]
Type=oneshot
ExecStartPre=/usr/bin/truncate -s 0 /var/www/changed
```

```
ExecStart=/usr/bin/chown -R www-data /var/www/html/
ExecStart=/usr/bin/chmod -R g+w /var/www/html/
ExecStart=/usr/bin/chmod -R o-r /var/www/html/
```

Erklärungen

Sektion [Service]:

"ExecStart="-Befehle werden nur ausgeführt, nachdem sich alle *"ExecStartPre="*-Befehle erfolgreich beendet haben. Zuerst wird die Datei `/var/www/changed` auf 0-Bite zurückgesetzt und danach der Rest ausgeführt.

7.12.4.1 Zusätzliche service-Unit anlegen Da die .path-Unit Verzeichnisse nicht rekursiv überwachen kann, benötigen wir für unser Beispiel eine zusätzliche service-Unit. Wir legen die Datei `server1-watch.service` im Verzeichnis `/usr/local/lib/systemd/system/` mit folgendem Inhalt an.

```
[Unit]
Description=Watching server1 folder.
Before=server1.path
Wants=server1.path

[Service]
Type=forking
ExecStart=inotifywait -dqr -e move,create -o /var/www/changed
    /var/www/html/

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

Anmerkung:

Interessant ist, dass systemd intern das inotify-API für path-Unit verwendet, um Dateisysteme zu überwachen, jedoch deren Rekursiv-Funktion nicht implementiert.

Erklärungen

Die Sektion [Unit]:

"Before=" und *"Wants="* sind die entsprechenden Korrelationen zu *"BindsTo="* und *"After="* aus der server1.service-Unit.

Sektion [Service]:
"inotifywait" protokolliert in die Datei `/var/www/changed`, die außerhalb von `DocumentRoot` des Apache-Webservers liegt.

7.12.5 path-Unit eingliedern

Auf Grund der Abhängigkeit gliedern wir zuerst die server1.path-Unit und dann die server1-watch.service-Unit in systemd ein. Die server1.service-Unit benötigt und beinhaltet keine [Install]-Sektion. Bei dem Versuch sie einzugliedern erhielten wir eine Fehlermeldung.

```
# systemctl enable server1.path
Created symlink /etc/systemd/system/multi-user.target.wants/⇙
   server1.path /usr/local/lib/systemd/system/server1.path.

# systemctl enable server1-watch.service
Created symlink /etc/systemd/system/multi-user.target.wants/⇙
   server1-watch.service /usr/local/lib/systemd/system/⇙
   server1-watch.service.
```

Nun ist das Monitoring auch gleich aktiv, wie uns die Statusausgaben aller drei Units zeigen.

```
# systemctl status server1-watch.service●
 server1-watch.service - Watching server1 folder.
     Loaded: loaded (/usr/local/lib/systemd/system/server1-⇙
        watch.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2021-02-21 19:25:20 ⇙
        CET; 1min 49s ago
    Process: 23788 ExecStart=inotifywait -dqr -e move,create ⇙
       -o /var/www/changed /var/www/html/ (code=exited, ⇙
       status=0/SUCCESS)
   Main PID: 23790 (inotifywait)
      Tasks: 1 (limit: 2322)
     Memory: 216.0K
        CPU: 5ms
     CGroup: /system.slice/server1-watch.service└─
             23790 inotifywait -dqr -e move,create -o /var/⇙
                www/changed /var/www/html/
```

```
Feb 21 19:25:20 lap1 systemd[1]: Starting Watching server1 ⇙
   folder....
Feb 21 19:25:20 lap1 systemd[1]: Started Watching server1 ⇙
   folder..

# systemctl status server1.path●
 server1.path - Monitoring "changed" file!
     Loaded: loaded (/usr/local/lib/systemd/system/server1.⇙
        path; enabled; vendor preset: enabled)
     Active: active (waiting) since Sun 2021-02-21 19:25:20 ⇙
        CET; 3min 27s ago
   Triggers: ● server1.service

Feb 21 19:25:20 lap1 systemd[1]: Started Monitoring "changed"⇙
    file!.

# systemctl status server1.service●
 server1.service - Change permissions in server1 folder
     Loaded: loaded (/usr/local/lib/systemd/system/server1.⇙
        service; static)
     Active: inactive (dead)
TriggeredBy: ● server1.path
```

Der Status *“Active: inactive (dead)”* der letzten Ausgabe ist der normale Zustand für die Unit server1.service, denn diese Unit ist nur dann aktiv, wenn sie von server1.path angestoßen wurde ihre Befehlskette auszuführen. Danach geht sie wieder in den inaktiven Zustand über.

7.12.6 service-Unit manuell ausführen

Sollte es einmal hilfreich oder nötig sein die Dateirechte in *DocumentRoot* des Apache-Webservers manuell zu ändern, setzen wir einfach diesen Befehl ab:

```
# systemctl start server1.service
```

Eine erneute Statusabfrage generiert zusätzlich einige Protokollzeilen, denen wir den erfolgreichen Durchlauf der Befehlskette entnehmen können.

```
# systemctl status server1.service●
 server1.service - Change permissions in server1 folder
     Loaded: loaded (/usr/local/lib/systemd/system/server1.↙
        service; static)
     Active: inactive (dead) since Mon 2021-02-22 17:55:36 ↙
        CET; 1min 43s ago
TriggeredBy: ● server1.path
    Process: 2822 ExecStartPre=truncate -s 0 /var/www/changed↙
        (code=exited, status=0/SUCCESS)
    Process: 2823 ExecStart=chown -R www-data /var/www/html1/↙
        (code=exited, status=0/SUCCESS)
    Process: 2824 ExecStart=chmod -R g+w /var/www/html1/ (↙
       code=exited, status=0/SUCCESS)
    Process: 2825 ExecStart=chmod -R o-r /var/www/html1/ (↙
       code=exited, status=0/SUCCESS)
   Main PID: 2825 (code=exited, status=0/SUCCESS)
        CPU: 19ms

Feb 22 17:55:36 lap1 systemd[1]: Starting Change permissions ↙
   in server1 folder...
Feb 22 17:55:36 lap1 systemd[1]: server1.service: Succeeded.
Feb 22 17:55:36 lap1 systemd[1]: Finished Change permissions ↙
   in server1 folder.
```

7.12.7 Quellen systemd-path

Deutsche Manpage ‘systemd.path’

Ein anders gelagertes Beispiel:
PRO-LINUX.DE, Systemd Path Units…

Seite zuletzt aktualisiert 2021-11-29

7.13 systemd-timer

Die grundlegenden und einführenden Informationen zu Systemd enthält die Handbuchseite Systemd-Start Die alle Unit-Dateien betreffenden Sektionen *[Unit]* und *[Install]* behandelt unsere Handbuchseite Systemd Unit-Datei
In der vorliegenden Handbuchseite erklären wir die Funktion der Unit **systemd.timer**, mit der zeitgesteuert Aktionen ausgelöst werden können.

Die *timer*-Unit wird meist eingesetzt, um regelmäßig anfallende Aktionen zu erledigen. Dazu ist eine gleichnamige *service*-Unit notwendig, in der die Aktionen definiert sind. Sobald der Systemzeitgeber mit der in der *timer*-Unit definierten Zeit übereinstimmt, aktiviert sie die gleichnamige *service*-Unit.
Bei entsprechender Konfiguration können verpasste Läufe, während die Maschine ausgeschaltet war, nachgeholt werden.
Auch ist es möglich, dass eine *timer*-Unit die gewünschten Aktionen nur ein einziges Mal zu einem vorher definierten Termin auslöst.

7.13.1 Benötigte Dateien

Die **systemd-timer**-Unit benötigt zwei Dateien mit dem gleichen Basename im Verzeichnis `/usr/local/lib/systemd/system/` für ihre Funktion. (Ggf. ist das Verzeichnis zuvor mit dem Befehl **`mkdir -p /usr/local/lib/systemd/system/`**↙ anzulegen.) Das sind die

- Timer-Unit-Datei (xxxxx.timer), welche die Zeitsteuerung und den Auslöser für die service-Unit enthält
  und die

- service-Unit Datei (xxxxx.service), welche die zu startende Aktion enthält.

Für umfangreichere Aktionen erstellt man als dritte Datei ein Skript in `/usr/local/bin/`↙, das von der service-Unit ausgeführt wird.

Wir erstellen in dem Beispiel ein regelmäßiges Backup mit `rsync`.

7.13.2 service-Unit für timer

Die service-Unit, die das Backup ausführt, wird von der timer-Unit aktiviert und kontrolliert und benötigt daher keine [Install] Sektion. Somit reicht die Beschreibung

der Unit in der Sektion [Unit]. Ihre Sektion [Service] enthält den auszuführenden Befehl nach der Option `ExecStart=`.

Wir legen die Datei `backup.service` im Verzeichnis `/usr/local/lib/systemd/system/` mit folgendem Inhalt an.

```
[Unit]
Description="Command to backup my home directory"

[Service]
Type=oneshot
ExecStart=/usr/bin/rsync -a --exclude=.cache/* /home/<user> /mnt/sdb5/backup/home/
```

Den String <user> bitte durch den eigenen User ersetzen.

7.13.3 timer-Unit anlegen

Wir legen die Datei `backup.timer` im Verzeichnis `/usr/local/lib/systemd/system/` mit folgendem Inhalt an.

```
[Unit]
Description="Backup my home directory"

[Timer]
OnCalendar=*-*-* 19:00:00
Persistent=true

[Install]
WantedBy=timers.target
```

Erklärungen
Die timer-Unit muss zwingend die Sektion [Timer] enthalten, in der festgelegt wird wann und wie die zugehörige service-Unit ausgelöst wird.
Es stehen zwei Timer-Typen zur Verfügung:

1. Realtime timers,
   die mit der Option `OnCalendar=` einen Echtzeit- (d.h. Wanduhr-)Zeitgeber definiert

(das Beispiel *“OnCalendar=*-*-* 19:00:00”* bedeutet “täglich um 19:00 Uhr”), und

2. Monotonic timers,
   die mit den Optionen `OnActiveSec=`, `OnBootSec=`, `OnStartupSec=`, `OnUnitActiveSec=`, `OnUnitInactiveSec=` einen zu der Option relativen Zeitgeber definiert.
   “OnBootSec=90” bedeutet “90 Sekunden nach dem Booten” und
   “OnUnitActiveSec=1d” bedeutet “Einen Tag nachdem der Zeitgeber letztmalig aktiviert wurde”.
   Beide Optionen zusammen lösen die zugehörige service-Unit 90 Sekunden nach den Booten und dann genau im 24 Stunden-Takt aus, solange die Maschine nicht heruntergefahren wird.

Die im Beispiel enthaltene Option `Persistent=` speichert den Zeitpunkt, zu dem die service-Unit das letzte Mal ausgelöst wurde, als leere Datei im Verzeichnis `/var/lib/systemd/timers/`. Dies ist nützlich, um verpasste Läufe, als die Maschine ausgeschaltet war, nachzuholen.

timer-Unit eingliedern

Mit dem folgenden Befehl gliedern wir die timer-Unit in systemd ein.

```
# systemctl enable backup.timer
Created symlink /etc/systemd/system/timers.target.wants/
   backup.timer \→
   /usr/local/lib/systemd/system/backup.timer.
```

Der analoge Befehl für die service-Unit ist nicht notwendig und würde auch zu einem Fehler führen, da in ihr keine [Install] Sektion enthalten ist.

timer-Unit manuell auslösen

Es wird nicht die timer-Unit, sondern die von ihr auszulösende service-Unit aufgerufen.

```
# systemctl start backup.service
```

7.13.4 timer-Unit als cron Ersatz

“cron” und “anacron” sind die bekanntesten und weit verbreiteten Job-Zeitplaner. Systemd Timer können eine Alternative sein. Wir betrachten kurz den Nutzen von, und die Vorbehalte gegen Systemd Timer.

Nutzen

- Jobs können Abhängigkeiten haben (von anderen systemd-Diensten abhängen).
- Timer Units werden im systemd-Journal geloggt.
- Man kann einen Job sehr einfach unabhängig von seinem Timer aufrufen.
- Man kann Timer Units einen Nice-Wert geben oder cgroups für die Ressourcenverwaltung nutzen.
- Systemd Timer Units können von Ereignissen wie dem Booten oder Hardware-Änderungen ausgelöst werden.
- Sie können auf einfache Weise mit systemctl aktiviert oder deaktiviert werden.

Vorbehalte

- Die Konfiguration eines Cron-Jobs ist ein einfacher Vorgang.
- Cron kann E-Mails mit Hilfe der MAILTO-Variablen senden.

7.13.5 Quellen systemd-timer

Deutsche Manpage ‘systemd.timer’
Archlinux Wiki, Timers
PRO-LINUX.DE, Systemd Timer Units…

Seite zuletzt aktualisiert 2021-11-29

7.14 Systemjournal

Das Systemjournal besteht aus dem *systemd-journald*, kurz **journald**, der Protokollmeldungen sammelt und speichert, und dem **journalctl**, das der Verwaltung, Abfrage und Ausgabe der gesammelten Protokollmeldungen dient.

7.14.1 journald

journald ist ein Systemdienst, der mit Hilfe der Unit *systemd-journald.service* (und seiner zugehörigen Socket-Units) Protokollmeldungen sammelt und speichert.
Es erstellt und verwaltet strukturierte, indizierte Journale, basierend auf den Protokollmeldungen aus:

- Kernel-Protokollmeldungen
- Einfache System-Protokollmeldungen
- Strukturierte System-Protokollmeldungen über die native Journal-API
- Standardausgabe und Standardfehlerausgabe der Dienste-Units
- Audit-Aufzeichnungen, stammend aus dem Kernel-Audit-Subsystem

journald erlaubt Journal-“Namensräume” (namespaces). Sie sind zum Einen ein Mechanismus zur logischen Isolation eines Protokoll-Datenstroms vom Rest des Systems, zum Anderen auch ein Mechanismus zur Steigerung der Leistung. Journal-Namensräume existieren gleichzeitig und nebeneinander. Jeder hat seinen eigenen, unabhängigen Protokolldatenstrom. Nach der Installation von siduction besteht nur der Vorgabe-Namensraum des Systems.

Der *journald* speichert die Protokolldaten standardmäßig dauerhaft unter `/var/log/journal/MASCHINENKENNUNG`.

Protokolldaten für andere Namensräume befinden sich in `/var/log/journal/MASCHINENKENNUNG.NAMENSRAUM`.

Der Befehl **`systemd-cat`** bietet zwei Möglichkeiten um unabhängig von systemd-Units Daten eines Prozesses an das Journal weiterzugeben.

1. **`systemd-cat <Programm> <Option(en)>`**
   Mit einem Programmaufruf oder Befehl verwendet, leitet *systemd-cat* alle Standardeingaben, Standardausgaben und Standardfehlerausgaben eines Prozesses zum Journal um.

2. **In einer Pipe verwendet,**
   dient *systemd-cat* als Filterwerkzeug, um die zuvor erstellte Ausgabe an das Journal zu senden.

Falls kein Parameter übergeben wurde, wird *systemd-cat* alles, was es von der Standardeingabe liest, an das Journal schicken. Die man-page systemd-cat.1.de bietet weitere Informationen.

7.14.2 journald über das Netzwerk

Die *systemd-journal*-Module *upload*, *remote* und *gatewayd* ermöglichen das Versenden und Empfangen von System-Protokolldaten zwischen verschiedenen Rechnern über das Netzwerk. Mit ihrer Hilfe lassen sich entfernte Rechner fortlaufend überwachen. In dieser Konstellation ist es sinnvoll auf dem Remoterechner Namensräume für die Protokolldaten der entfernten Rechner einzurichten.
Für weitere Informationen bitte die man-pages journal upload, journal remote und journal gatewayd lesen.

7.14.3 journald.conf

Die folgenden Dateien konfigurieren verschiedene Parameter des systemd-Journal-Dienstes.

- /etc/systemd/journald.conf
- /etc/systemd/journald.conf.d/*.conf
- /etc/systemd/journald@NAMENSRAUM.conf (optional)
- /run/systemd/journald.conf.d/*.conf (optional)
- /usr/lib/systemd/journald.conf.d/*.conf (optional)

Der Vorgabe-Namensraum, den der systemd-journald.service (und seine zugehörigen Socket-Units) verwaltet, wird in `/etc/systemd/journald.conf` und zugeordneten Ergänzungen konfiguriert.
Die Konfigurationsdatei enthält die Vorgaben als auskommentierten Hinweis für

den Administrator. Um lokal Einstellungen zu ändern, genügt es diese Datei zu bearbeiten.

Instanzen, die andere Namensräume verwalten, werden nur benötigt, wenn von den Vorgaben abgewichen werden muss. Deren Konfigurationsdatei ist nach dem Muster `etc/systemd/journald@NAMENSRAUM.conf` zu erstellen.
Einem bestimmten Journal-Namensraum können Dienste-Units mittels der Unit-Dateieinstellung `LogNamespace=` zugeordnet werden.

Standardmäßig sammelt nur der Vorgabe-Namensraum Kernel- und Auditprotokollnachrichten.

Rangfolge

Wenn Pakete die Konfiguration anpassen müssen, können sie Konfigurationsschnipsel in `/usr/lib/systemd/*.conf.d/` oder `/usr/local/lib/systemd/*.conf.d/` installieren.

Die Hauptkonfigurationsdatei wird vor jeder anderen aus den Konfigurationsverzeichnissen gelesen und hat die niedrigste Priorität. Einträge in einer Datei in jedem der Konfigurationsverzeichnisse setzen Einträge in der Hauptkonfigurationsdatei außer Kraft. Dateien in den Unterverzeichnissen `*.conf.d/` werden nach ihrem Dateinamen sortiert, unabhängig davon, in welchem Unterverzeichnis sie sich befinden. Sofern eigene Konfigurationsdateien nötig sind, wird empfohlen, allen Dateinamen in diesen Unterverzeichnissen eine zweistellige Zahl und einen Bindestrich voranzustellen, um die Sortierung der Dateien zu vereinfachen.

7.14.4 journalctl

journalctl dient der Abfrage des von systemd-journald erstellten Journals.
Beim Aufruf ohne Parameter wird der gesamte Inhalt aus allen zugreifbaren Quellen des Journals angezeigt, beginnend mit dem ältesten Eintrag.
Die Ausgabe wird seitenweise durch *less* geleitet. Lange Zeilen kann man mittels der **`Pfeil-links`** und **`Pfeil-rechts`** Tasten betrachten. Die Option `--no-pager` deaktiviert die seitenweise Anzeige, wobei die Zeilen auf die Breite des Terminals verkürzt werden.

journalctl bietet zu den nachfolgend beschriebenen Optionen eine ganze Reihe weiterer Möglichkeiten der Filterung und Aufbereitung der Ausgaben. Bitte auch die man-Page journalctl, Journalabfrage lesen.

Rechte

Dem Benutzer **root** und allen Benutzern die Mitglied der Gruppen **systemd-journal**, **adm** und **wheel** sind, wird Zugriff auf das System-Journal und die Journale der anderen Benutzer gewährt. Siduction fügt alle konfigurierten USER der Gruppe **systemd-journal** zu.

Das Journal enthält vertrauenswürdige Felder, d.h. Felder, die implizit vom Journal hinzugefügt werden und durch Client-Code nicht geändert werden können. Sie beginnen mit einem Unterstrich. (z.B.: _PID=, _UID=, _GID=, _COMM=, _EXE=, _CMDLINE=)

Ausgabe filtern

- *Optionen*: `--user`, `--system`, `--directory=`, `--file=`, `--namespace=`
  Die Optionen begrenzen die Quelle der Ausgabe auf den genannten Bereich, das Verzeichnis oder die Datei.
- *Optionen*: `-b`, `-k`, `-u`, `-p`, `-g`, `-S`, `-U`
  Die Ausgaben dieser Optionen verwenden alle zu Verfügung stehenden Journal-Dateien, es sei denn, eine der zuvor genannten Optionen wird zusätzlich verwendet.
  - `-b --boot=`
    Zeigt Nachrichten von einem bestimmten Systemstart. Ohne Argument werden die Protokolle für den aktuellen Systemstart angezeigt. Das Argument “-1” gibt die Meldungen des Systemstarts vor dem Aktuellen aus. Das Argument “5” präsentiert die Meldungen des fünften Systemstarts seit Beginn der Aufzeichnungen.
  - `-k --dmesg`
    Zeigt nur Kernelnachrichten. Dies beinhaltet die Option `-b`, sodass nur Kernelmeldungen seit dem aktuellen Systemstart ausgegeben werden.
  - `-u --unit=`
    Diese Option benötigt die Angabe einer UNIT oder eines MUSTERs.
    Gibt die Journaleinträge für die angegebene systemd-Unit UNIT oder für alle Units, die auf das MUSTER passen, aus.
  - `-p --priority=`
    Filtert die Ausgabe nach Nachrichtenprioritäten oder Prioritätsberei-

chen. Benötigt die Angabe einer einzelnen Protokollstufe, oder einen Bereich von Protokollstufen in der Form VON..BIS.
Die Protokollstufen sind die normalen Syslog-Protokollstufen:
“emerg” (0), “alert” (1), “crit” (2), “err” (3), “warning” (4), “notice” (5), “info” (6), “debug” (7)
Als Argument können sowohl die Namen als auch die Ziffern der Protokollstufen verwendet werden. Falls eine einzelne Protokollstufe angegeben ist, werden alle Nachrichten mit dieser oder einer niedrigeren Protokollstufe angezeigt.

- `-g --grep=`
  Benötigt die Angabe eines PERL-kompatiblen regulären Ausdrucks, um die Ausgabe zu filtern. Der reguläre Ausdruck wird in den Journaleinträgen auf das Feld “MESSAGE=” angewendet.

- `-S --since=` und `-U --until=`
  Die Anzeige beginnt mit neueren Einträgen ab dem angegebenen Datum oder älteren Einträgen bis zum angegebenen Datum. Das Datumsformat sollte “2012-10-30 18:17:16” sein, es können aber auch Teile davon weggelassen werden. Alternativ sind die Zeichenketten “yesterday”, “today”, “tomorrow” möglich. Das Argument “now” bezieht sich auf die aktuelle Zeit. Die Angabe relativer Zeiten ermöglichen ein vorangestelltes “-” oder “+”, die sich auf Zeiten vor bzw. nach der angegebenen Zeit beziehen.

Ausgabe steuern

- Optionen: `-f`, `-n`, `-r`, `-o`, `-x`, `--no-pager`

  - `-f --follow`
    Nur die neusten Journal-Einträge anzeigen und kontinuierlich neue Einträge ausgeben. Dies beinhaltet die Option `-n`. Die Ausgabe ist vergleichbar mit dem altbekannten Befehl `tail -f /var/log/messages`.

  - `-n --lines=`
    Zeigt die neusten Journal-Einträge und begrenzt die Anzahl der zu zeigenden Ereignisse. Das Argument ist eine positive Ganzzahl. Der Vorgabewert ist 10, falls kein Argument angegeben wird.

- `-r --reverse`
  Die Ausgabe beginnt mit dem neusten Eintrag.

- `-o --output=`
  Steuert die Formatierung der angezeigten Journal-Einträge. Dieser Option sind eine ganze Reihe weiterer Optionen untergeordnet, von denen wir hier nur die Option “short-full” betrachten.

  `-o short-full`
  Die Ausgabe ist größtenteils identisch zu der Formatierung klassischer Syslog-Dateien. Sie zeigt eine Zeile pro Journal-Eintrag an, aber der Zeitstempel wird im Format, das die Optionen –since= und –until= akzeptieren, ausgegeben. Deshalb eignet sich diese Ausgabe sehr gut um nachfolgend eine zeitbezogene Filterung der Journaleinträge zu erstellen.

- `-x` - -catalog
  Ergänzt Protokollzeilen mit erklärenden Hilfetexten, soweit diese verfügbar sind.

- `--no-pager`
  Die Option deaktiviert die seitenweise Anzeige, wobei die Zeilen auf die Breite des Terminals verkürzt werden. Sie zu benutzen ist nur sinnvoll, wenn für die Ausgabe nur eine geringe Anzahl an Zeilen erwartet wird.

journalctl steuern

- Optionen: `--disk-usage`, `--vacuum-size=`, `--vacuum-time=`, `--vacuum-files`, `--rotate`, `--verify`
  Die Optionen behandeln die Verwaltung der von *journald* geschriebenen Daten.

  - `--disk-usage`
    Zeigt den aktuellen Plattenplatzverbrauch aller Journal-Dateien an.

  - `--vacuum-size=`, `--vacuum-time=`, `--vacuum-files`
    Entfernt die ältesten archivierten Journal-Dateien, bis der Plattenplatz, den sie verwenden, unter die angegebene Größe fällt oder alle archivierten Journal-Dateien, die keine Daten älter als die angegebene Zeitspanne enthalten oder so dass nicht mehr als die angegebene Anzahl an

separaten Journal-Dateien verbleiben. Die Ausführung von `--vacuum-xxx` bezieht nicht die aktiven Journal-Dateien ein.

- `--rotate`
  Bittet den Journal-Daemon, die Journal-Dateien zu rotieren. Journal-Dateien-Rotation hat den Effekt, dass alle derzeit aktiven Journal-Dateien als archiviert markiert und umbenannt werden, so dass in der Zukunft niemals mehr in sie geschrieben wird. Dann werden stattdessen neue (leere) Journal-Dateien erstellt. Diese Aktion kann mit `--vacuum-xxx` in einem einzigen Befehl kombiniert werden, um die `--vacuum-xxx` mitgegebenen Argumente tatsächlich zu erreichen.

- `--verify`
  Prüft die Journal-Dateien auf interne Konsistenz.

7.14.5 journalctl beherrschen

Wie oben unter Rechte beschrieben, kannst Du das Journal als einfacher User benutzen. Hier sind einige Beispiele:

Befehl	Anzeige
journalctl	das volle Journal aller User, älteste Einträge zuerst
journalctl -r	wie zuvor, neueste Einträge zuerst
journalctl -b	das Protokoll des letzten Bootvorgangs
journalctl -b -1 -k	vom vorletzten Bootvorgang (-1) alle Kernelmeldungen
journalctl -b -p err	limitiert auf den letzten Boot und die Priorität ERROR
journalctl –since=yesterday	das Journal seit gestern
journalctl /dev/sda	das Journal der Gerätedatei /dev/sda
journalctl /usr/bin/dbus-daemon	alle Logs des D-Bus-Daemon
journalctl -f	Liveansicht des Journal (früher: tail -f /var/log/messages)

Die Option "'–list-boots*" gibt die entsprechende Liste aus.

```
# journalctl --list-boots --no-pager
[...]
 -50 8fc07f387... Sun 2021-02-28 11:07:05 –CETSun 2021-02-28 ⇙
    23:01:56 CET
 -49 aa49cb3af... Mon 2021-03-01 17:49:58 –CETMon 2021-03-01 ⇙
    20:19:59 CET
 -48 3a6e55a4a... Tue 2021-03-02 12:18:46 –CETTue 2021-03-02 ⇙
    20:47:24 CET
 -47 a46150a19... Wed 2021-03-03 11:06:29 –CETWed 2021-03-03 ⇙
    20:33:09 CET
 -46 d42ed8b05... Thu 2021-03-04 10:59:56 –CETThu 2021-03-04 ⇙
    19:53:26 CET
 -45 566f65991... Thu 2021-03-04 19:53:52 –CETThu 2021-03-04 ⇙
    19:55:38 CET
 -44 8e2da4a61... Fri 2021-03-05 10:15:18 –CETFri 2021-03-05 ⇙
    10:20:11 CET
[...]
```

Anschließend kannst du dir mit dem Befehl **`journalctl -b -47`** die Meldungen des Bootvorgangs vom 3.3.2021 anzeigen lassen.

Eine weitere Neuerung beim Protokollieren ist die Tab-Vervollständigung für journalctl. Wenn Du *journalctl* schreibst, und zwei mal die **`TAB`** Taste drückst, erscheint eine Liste möglicher Vervollständigungen:

```
$ journalctl
_AUDIT_FIELD_APPARMOR=        _KERNEL_SUBSYSTEM=
_AUDIT_FIELD_CAPABILITY=      KERNEL_USEC=
_AUDIT_FIELD_CAPNAME=         LEADER=
_AUDIT_FIELD_DENIED_MASK=     LIMIT=
_AUDIT_FIELD_INFO=            LIMIT_PRETTY=
_AUDIT_FIELD_NAME=            _LINE_BREAK=
_AUDIT_FIELD_OPERATION=       _MACHINE_ID=
_AUDIT_FIELD_OUID=            MAX_USE=
_AUDIT_FIELD_PEER=            MAX_USE_PRETTY=
_AUDIT_FIELD_PROFILE=         MESSAGE=
_AUDIT_FIELD_REQUESTED_MASK=  MESSAGE_ID=
_AUDIT_FIELD_SIGNAL=          NM_CONNECTION=
```

```
_AUDIT_ID=                      NM_DEVICE=
_AUDIT_LOGINUID=                NM_LOG_DOMAINS=
_AUDIT_SESSION=                 NM_LOG_LEVEL=
_AUDIT_TYPE=                    N_RESTARTS=
_AUDIT_TYPE_NAME=               _PID=
AVAILABLE=                      PRIORITY=
AVAILABLE_PRETTY=               SEAT_ID=
_BOOT_ID=                       _SELINUX_CONTEXT=
_CAP_EFFECTIVE=                 SESSION_ID=
_CMDLINE=                       SHUTDOWN=
CODE_FILE=                      SLEEP=
CODE_FUNC=                      _SOURCE_MONOTONIC_TIMESTAMP=
CODE_LINE=                      _SOURCE_REALTIME_TIMESTAMP=
_COMM=                          _STREAM_ID=
COMMAND=                        SYSLOG_FACILITY=
CONFIG_FILE=                    SYSLOG_IDENTIFIER=
CONFIG_LINE=                    SYSLOG_PID=
CURRENT_USE=                    SYSLOG_RAW=
CURRENT_USE_PRETTY=             SYSLOG_TIMESTAMP=
DISK_AVAILABLE=                 _SYSTEMD_CGROUP=
DISK_AVAILABLE_PRETTY=          _SYSTEMD_INVOCATION_ID=
DISK_KEEP_FREE=                 _SYSTEMD_OWNER_UID=
DISK_KEEP_FREE_PRETTY=          _SYSTEMD_SESSION=
ERRNO=                          _SYSTEMD_SLICE=
_EXE=                           _SYSTEMD_UNIT=
EXECUTABLE=                     _SYSTEMD_USER_SLICE=
EXIT_CODE=                      _SYSTEMD_USER_UNIT=
EXIT_STATUS=                    THREAD_ID=
_FSUID=                         TIMESTAMP_BOOTTIME=
_GID=                           TIMESTAMP_MONOTONIC=
GLIB_DOMAIN=                    _TRANSPORT=
GLIB_OLD_LOG_API=               _UDEV_DEVNODE=
_HOSTNAME=                      _UDEV_SYSNAME=
INVOCATION_ID=                  _UID=
JOB_ID=                         UNIT=
JOB_RESULT=                     UNIT_RESULT=
JOB_TYPE=                       USER_ID=
JOURNAL_NAME=                   USER_INVOCATION_ID=
```

```
JOURNAL_PATH=                   USERSPACE_USEC=
_KERNEL_DEVICE=                 USER_UNIT=
```

Die meisten davon sind selbsterklärend. Beispielsweise COMM, was für *command* steht, bedient eine Menge an Optionen:

`journalctl _COMM=` listet nach einem weiterer Druck auf TAB die möglichen Applikationen:

```
$ journalctl _COMM=
acpid           hddtemp        ntpdate       systemd
acpi-fakekey    hdparm         ntpd          systemd-fsck
acpi-support    hp             ofono         systemd-hostnam
alsactl         hpfax          ofonod        systemd-journal
anacron         ifup           pkexec        systemd-logind
apache2         irqbalance     polkitd       systemd-modules
backlighthelper kbd            pulseaudio    systemd-shutdow
bash            kdm            pywwetha      systemd-udevd
bluetoothd      keyboard-setup pywwetha.py   teamviewerd
chfn            loadcpufreq    resolvconf    udev-configure-
chrome          logger         rpcbind       udisksd
console-kit-dae login          rpc.statd     udisks-daemon
console-setup   lvm            samba-ad-dc   umount
cpufrequtils    lvm2           saned         uptimed
cron            mbd            sensors       useradd
cups            mbmon          sh            usermod
dbus-daemon     mdadm          smartmontools vboxdrv
ddclient        mdadm-raid     smbd          VBoxExtPackHelp
docvert-convert mtp-probe      ssh           vdr
glances         mysql          sshd          winbind
gpasswd         networking     su
gpm             nfs-common     sudo
groupadd        ntp            sysstat
```

Mit **`journalctl _COMM=su`** kannst du nun sehen, welcher User sich wann mit “su” root-Rechte verschafft hat.

```
# journalctl _COMM=su
-- Boot 1b5d2b3fcd9043d88d8abce665b75ed4 --
```

```
Mär 10 13:45:53 pc1 su[75259]: (to root) siduser on pts/3
Mär 10 13:45:53 pc1 su[75259]: pam_unix(su:session): session ↙
   opened for user root(uid=0) by (uid=1000)
Mär 10 16:27:22 pc1 su[105197]: (to root) siduser on pts/1
Mär 10 16:27:22 pc1 su[105197]: pam_unix(su:session): session↙
    opened for user root(uid=0) by (uid=1000)
Mär 10 17:54:33 pc1 su[105197]: pam_unix(su:session): session↙
    closed for user root
Mär 10 17:54:42 pc1 su[75259]: pam_unix(su:session): session ↙
   closed for user root
-- Boot 37b19f6321814620be1ed4deb3be467f --
Mär 10 17:56:35 pc1 su[3381]: (to root) siduser on pts/1
Mär 10 17:56:35 pc1 su[3381]: pam_unix(su:session): session ↙
   opened for user root(uid=0) by (uid=1000)
Mär 10 19:07:17 pc1 su[3381]: pam_unix(su:session): session ↙
   closed for user root
```

Ein anderes Beispiel:

Man kann die Ausgabe zusätzlich zeitlich eingrenzen.

```
# journalctl _COMM=dbus-daemon --since=2020-04-06 --until↙
   ="2020-04-07 23:40:00"
[...]
Apr 07 22:59:04 pc1 org.gtk.Private.GPhoto2VolumeMonitor↙
   [2006]: ### debug: in handle_supported
Apr 07 22:59:04 pc1 org.gtk.Private.GPhoto2VolumeMonitor↙
   [2006]: ### debug: in handle_list
Apr 07 22:59:04 pc1 org.gtk.Private.GoaVolumeMonitor[2006]: #↙
   ## debug: in handle_supported
Apr 07 22:59:04 pc1 org.gtk.Private.GoaVolumeMonitor[2006]: #↙
   ## debug: in handle_list
Apr 07 23:03:09 pc1 org.gtk.Private.GPhoto2VolumeMonitor↙
   [2006]: ### debug: Name owner ':1.4320' vanished
Apr 07 23:03:09 pc1 org.gtk.Private.GoaVolumeMonitor[2006]: #↙
   ## debug: Name owner ':1.4320' vanished
Apr 07 23:03:09 pc1 org.gtk.Private.AfcVolumeMonitor[2006]: #↙
   ## debug: Name owner ':1.4320' vanished
Apr 07 23:03:09 pc1 org.gtk.Private.MTPVolumeMonitor[2006]: #↙
   ## debug: Name owner ':1.4320' vanished
```

Viele der oben genannten Optionen lassen sich miteinander kombinieren, damit nur die gesuchten Journaleinträge angezeigt werden. Die man-page von journalctl beschreibt alle Optionen ausführlich.

7.14.6 Quellen journald

systemd-journald
journald Konfiguration
journalctl, Journalabfrage
journal gatewayd
journal remote
journal upload
systemd-cat.1.de

Dank an Helge Kreuzmann für die deutschen Übersetzungen.

Seite zuletzt aktualisiert 2021-11-29

7.15 Testpage for people with color vision deficiency

7.15.1 Warning

Caution
While logged into the terminal with root privileges…

7.15.2 Code block

The option `--list-boots` prints the corresponding list.

- Liste

```
[...]
-50 8fc07f387... Sun 2021-02-28 11:07:05
-49 aa49cb3af... Mon 2021-03-01 17:49:58
1--------1---------2---------3---------4---------5 - - -

1--------1---------2---------3---------4---------5- - -
[...]
```

7.15.3 Inline code

The tilde `~` shows that you are in your home directory `/home/username`, and the dollar sign (the prompt) `$` means you are logged in. With the key combination **ALT** +**F2** or **F3** etc., you can switch from

7.15.4 Link

Read this man page: journal gatewayd
journal remote