



# Siduction Handbuch

siduction Team

03.03.2026

# Inhaltsverzeichnis

<b>1</b>	<b>Willkommen</b>	<b>11</b>
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	Release Notes 2024.1.0 Shine on ... . . . .	13
1.2.1	Was erwartet euch bei siduction 2024.1.0 . . . . .	13
1.2.2	siduction Editionen . . . . .	13
1.2.3	siduction-btrfs verbessert . . . . .	14
1.2.4	Non-free and Contrib . . . . .	16
1.2.5	Installationshinweise und bekannte Probleme . . . . .	17
1.3	Credits . . . . .	18
1.3.1	Credits für siduction 2024.1.0 . . . . .	18
1.3.2	Credits für das Handbuch . . . . .	18
1.3.3	Unser Dank gilt allen Beteiligten und unseren treuen Usern	19
1.4	siduction Hilfe . . . . .	20
1.4.1	Das siduction-Forum . . . . .	20
1.4.2	IRC - interaktiver Livesupport . . . . .	20
1.4.3	Nützliche Helfer im Textmodus . . . . .	21
1.4.4	siduction IRC-Support im Textmodus . . . . .	22
1.4.5	Surfen im Internet im Textmodus . . . . .	24
1.4.6	inxi . . . . .	25
1.4.7	Nützliche Links . . . . .	25
1.5	Skripte in siduction . . . . .	27
1.5.1	Vom Benutzer ausführbar . . . . .	27
1.5.2	Im Hintergrund laufend . . . . .	28
<b>2</b>	<b>Quickstart</b>	<b>29</b>
2.1	Einleitung . . . . .	30
2.1.1	Essenzielle Kapitel . . . . .	30
2.2	Kernel und Softwarepakete . . . . .	32
2.2.1	Die Verwaltung von Softwarepaketen . . . . .	32
2.2.2	Aktualisierung des Systems - upgrade . . . . .	33
2.3	Konfiguration von Netzwerken . . . . .	35

2.3.1	NetworkManager . . . . .	35
2.4	Schnellinstallation . . . . .	37
2.4.1	Fünf Schritte zum Ziel . . . . .	37
<b>3</b>	<b>ISO-Abbilder</b>	<b>38</b>
3.1	Inhalt der Live-ISO . . . . .	39
3.1.1	Hinweis zur Software auf dem Live-ISO . . . . .	39
3.1.2	Varianten der ISO . . . . .	39
3.1.3	Minimale Systemanforderungen . . . . .	40
3.1.4	Anwendungen und Hilfsprogramme . . . . .	41
3.1.5	Haftungsausschluss_Disclaimer . . . . .	41
3.2	Live-Medium verwenden . . . . .	42
3.2.1	Eingerichtete User auf dem Live-Medium . . . . .	42
3.2.2	Chroot Helfer . . . . .	42
3.2.3	Mit root-Rechten auf dem Live-Medium . . . . .	42
3.2.4	Ein neues Passwort setzen . . . . .	43
3.2.5	Software-Installation bei Live-Sitzung . . . . .	44
3.3	Bootoptionen Cheatcodes . . . . .	45
3.3.1	siduction spezifische Parameter . . . . .	45
3.3.2	Bootoptionen für den Grafikserver X . . . . .	47
3.3.3	Allgemeine Parameter des Linux-Kernels . . . . .	49
3.3.4	VGA-Codes . . . . .	50
3.4	ISO download . . . . .	52
3.4.1	Dateien der siduction-Spiegelserver . . . . .	54
3.4.2	Integritätsprüfung . . . . .	55
3.5	ISO auf USB-Stick - Speicherkarte . . . . .	57
3.5.1	GUI Anwendung . . . . .	57
3.5.2	Linux Kommandozeile . . . . .	58
3.5.2.1	Zusätzliche Datenpartition . . . . .	59
3.5.3	Mac OS X Kommandozeile . . . . .	60
3.6	ISO brennen . . . . .	61
3.6.1	DVD mit Linux brennen . . . . .	61
3.6.2	DVD mit Windows brennen . . . . .	62
3.7	Life-DVD ohne GUI brennen . . . . .	63
3.7.1	burniso . . . . .	63
3.7.2	Brennen mit cdrdao wodim growisofs . . . . .	64

3.7.3	Verfügbare Geräte . . . . .	64
3.7.4	Beispiele für CD DVD BD . . . . .	65
<b>4</b>	<b>Installation</b>	<b>68</b>
4.1	Installation auf HDD . . . . .	69
4.1.1	Datensicherung . . . . .	69
4.1.2	Installationsvorbereitungen . . . . .	69
4.1.3	Partitionierung und Dateisysteme . . . . .	70
4.1.4	Duplizierung auf einen anderen Computer . . . . .	70
4.1.5	Das Installationsprogramm Calamares . . . . .	71
4.1.6	System verschlüsseln . . . . .	77
4.1.7	Benutzer hinzufügen . . . . .	78
4.2	Aus ISO-Datei booten . . . . .	80
4.2.1	Überblick . . . . .	80
4.2.2	fromiso mit Grub2 . . . . .	80
4.2.3	toram . . . . .	82
4.3	Partitionierung von Installationsmedien . . . . .	83
4.3.1	Mindestanforderungen . . . . .	83
4.3.2	Beispiele mit verschiedenen Plattengrößen . . . . .	84
4.3.3	Dateisysteme der Partitionen . . . . .	86
4.3.4	Partitionierungsprogramme . . . . .	87
4.3.5	Weiterführende Infos . . . . .	88
4.4	UUID - Benennung von Blockgeräten . . . . .	89
4.4.1	Arten der Benennung von Blockgeräten . . . . .	89
4.4.2	Label verwenden . . . . .	90
4.5	Die fstab . . . . .	91
4.5.1	Anpassung der fstab . . . . .	92
4.5.2	Erstellung neuer Einhängepunkte . . . . .	93
4.6	Partitionieren mit GParted . . . . .	95
4.6.1	Wichtige Hinweise . . . . .	95
4.6.2	GParted verwenden . . . . .	95
4.6.3	fstab anpassen . . . . .	101
4.6.4	NTFS-Partitionsgrößen mit GParted ändern . . . . .	102
4.7	Partitionieren mit fdisk . . . . .	104
4.7.1	Informationen zu Speichergeräten . . . . .	104
4.7.2	Cfdisk verwenden . . . . .	107

4.7.3	Formatieren von Partitionen . . . . .	111
4.8	LVM-Partitionierung - Logical Volume Manager . . . . .	113
4.8.1	Sechs Schritte zu Logical Volumes . . . . .	114
4.8.2	Größenänderung eines Volumens . . . . .	116
4.8.3	LVM mit einem GUI-Programm verwalten . . . . .	118
4.8.4	Weitere Infos . . . . .	118
4.9	Das Verzeichnis home verschieben . . . . .	120
4.9.1	Private Daten verschieben . . . . .	121
4.9.2	fstab anpassen . . . . .	125
<b>5</b>	<b>Netzwerk</b>	<b>126</b>
5.1	Network Manager Kommandline Tool . . . . .	127
5.1.1	Network Manager verwenden . . . . .	127
5.1.2	Weiterführende Informationen . . . . .	130
5.2	IWD . . . . .	132
5.2.1	Graphische Konfigurationsprogramme . . . . .	132
5.2.2	Konfiguration im Terminal . . . . .	132
5.3	IWD statt wpa_supplicant . . . . .	135
5.3.1	IWD installieren . . . . .	135
5.3.2	Zurück zum wpa_supplicant . . . . .	138
5.4	SAMBA . . . . .	140
5.4.1	Client-Konfiguration . . . . .	140
5.4.2	siduction als Samba-Server . . . . .	141
5.5	SSH . . . . .	142
5.5.1	SSH absichern . . . . .	142
5.5.2	SSH für X-Window Programme . . . . .	145
5.5.3	Kopieren scp via ssh . . . . .	146
5.5.4	SSH mit Dolphin oder Thunar . . . . .	147
5.5.5	SSHFS - auf einem entfernten Computer mounten . . . . .	148
5.6	LAMP-Webserver . . . . .	151
5.6.1	Apache installieren . . . . .	153
5.6.2	MariaDB installieren . . . . .	155
5.6.3	PHP installieren . . . . .	155
5.6.4	phpMyAdmin installieren . . . . .	157
5.6.5	Weitere Software . . . . .	158
5.6.6	Statusangaben Log-Dateien . . . . .	158

5.6.7	Troubleshooting . . . . .	160
5.6.7.1	Wenn nichts hilft . . . . .	162
5.6.8	Sicherheit . . . . .	163
5.7	Apache einrichten . . . . .	165
5.7.1	Apache im Dateisystem . . . . .	165
5.7.2	Verbindung zum Server . . . . .	165
5.7.3	Apache Konfiguration . . . . .	167
5.7.4	Benutzer und Rechte . . . . .	169
5.7.5	Sicherheit - Apache Standard . . . . .	171
5.7.6	Sicherheit - weitere Konfigurationen . . . . .	172
5.7.7	HTTPS verwenden . . . . .	174
5.7.8	Sicherheits Tipps . . . . .	175
5.7.9	Integration in Apache2 . . . . .	175
5.7.10	Quellen Apache . . . . .	177
5.8	MariaDB einrichten . . . . .	178
5.8.1	MariaDB im Dateisystem . . . . .	178
5.8.2	Erstkonfiguration . . . . .	178
5.8.3	MariaDB CLI . . . . .	180
5.8.4	phpMyAdmin . . . . .	183
5.8.5	Integration in Systemd . . . . .	188
5.8.6	MariaDB Log . . . . .	188
5.8.7	Quellen MariaDB . . . . .	189
5.9	PHP einrichten . . . . .	190
5.9.1	PHP im Dateisystem . . . . .	190
5.9.2	PHP-Unterstützung für Apache2 . . . . .	190
5.9.3	PHP Konfiguration . . . . .	191
5.9.4	PHP Module . . . . .	191
5.9.5	Apache Log . . . . .	194
5.9.6	Quellen PHP . . . . .	195
<b>6</b>	<b>Hardware</b>	<b>196</b>
6.1	Grafiktreiber . . . . .	196
6.1.1	Open Source Xorg-Treiber . . . . .	196
6.1.2	Proprietäre Treiber . . . . .	197
6.1.3	Videotreiber 2D . . . . .	197
6.1.4	Videotreiber 3D . . . . .	197

6.1.5	nVidia closed Source Treiber . . . . .	198
<b>7</b>	<b>Systemadministration</b>	<b>201</b>
7.1	Terminal - Kommandozeile . . . . .	203
7.1.1	Arbeit als root . . . . .	204
7.1.2	Farbiges Terminal . . . . .	206
7.1.3	Wenn das Terminal hängt . . . . .	209
7.1.4	Hilfe im Terminal . . . . .	209
7.1.5	Linux Konsolenbefehle . . . . .	210
7.1.6	Skripte benutzen . . . . .	211
7.2	Systemadministration allgemein . . . . .	213
7.2.1	Bootoptionen Cheatcodes . . . . .	213
7.2.2	systemd - Dienste verwalten . . . . .	213
7.2.3	systemd.service . . . . .	214
7.2.4	systemd - UNIT eingliedern . . . . .	215
7.2.5	systemd-target ehemals Runlevel . . . . .	215
7.2.6	Beenden eines Prozesses . . . . .	217
7.2.7	Vergessenes Rootpasswort . . . . .	218
7.2.8	Setzen neuer Passwörter . . . . .	218
7.2.9	Schriftarten in siduction . . . . .	219
7.2.10	Userkonfiguration . . . . .	221
7.2.11	CUPS - das Drucksystem . . . . .	223
7.2.12	Sound in siduction . . . . .	224
7.3	Doas - Alternative zu Sudo . . . . .	226
7.3.1	Doas konfigurieren . . . . .	226
7.3.2	Doas und mehrere Benutzer . . . . .	227
7.4	Btrfs . . . . .	230
7.4.1	Btrfs Subvolumen . . . . .	231
7.4.2	Btrfs Snapshot . . . . .	234
7.5	Snapper . . . . .	236
7.5.1	Snapper Konfiguration . . . . .	236
7.5.2	Snapper und systemd . . . . .	239
7.5.3	Snapper - manuelle Snapshots . . . . .	241
7.5.4	Snapper Rollback . . . . .	244
7.5.5	Datei Rollback im Root-Dateisystem . . . . .	245
7.5.6	Datei Rollback von User Daten . . . . .	247

7.5.7	Quellen BTRFS und Snapper . . . . .	250
7.6	APT Paketverwaltung . . . . .	252
7.6.1	sources.list.d - Liste der Quellen . . . . .	252
7.6.2	apt und apt-get . . . . .	254
7.6.3	apt update . . . . .	256
7.6.4	Pakete installieren . . . . .	256
7.6.5	Pakete entfernen . . . . .	258
7.6.6	Hold oder Downgraden eines Pakets . . . . .	259
7.6.7	Aktualisierung des Systems . . . . .	261
7.6.8	Aktualisierbare Pakete . . . . .	262
7.6.9	full-upgrade ausführen . . . . .	264
7.6.10	Warum ausschließlich apt verwenden . . . . .	265
7.6.11	Programmpakete suchen . . . . .	266
7.7	Lokaler APT-Mirror . . . . .	270
7.7.1	Server installieren . . . . .	270
7.7.2	Client Konfiguration . . . . .	273
7.8	Nala Paketverwaltung . . . . .	275
7.8.1	Nala verwenden . . . . .	275
7.8.2	Befehle analog zu APT . . . . .	276
7.8.3	Befehle die APT nicht enthält . . . . .	277
7.9	Kernel Upgrade . . . . .	280
7.9.1	Kernel-Aktualisierung ohne Systemaktualisierung . . . . .	280
7.9.2	Module . . . . .	280
7.9.3	Entfernen alter Kernel . . . . .	280
7.10	Systemd der System- und Dienste-Manager . . . . .	282
7.10.1	Konzeption des systemd . . . . .	282
7.10.2	Unit Typen . . . . .	283
7.10.3	Systemd im Dateisystem . . . . .	284
7.10.4	Weitere Funktionen von systemd . . . . .	284
7.10.5	Handhabung von Diensten . . . . .	284
7.10.6	Quellen systemd . . . . .	286
7.11	systemd unit-Datei . . . . .	287
7.11.1	Ladepfad der Unit-Dateien . . . . .	287
7.11.2	Aktivierung der Unit-Datei . . . . .	288
7.11.3	Sektionen der Unit-Datei . . . . .	289
7.11.3.1	Sektion Unit . . . . .	289

7.11.3.2 Typ-spezifische Sektion . . . . .	293
7.11.3.3 Sektion Install . . . . .	293
7.11.4 Beispiel cupsd . . . . .	295
7.11.5 Werkzeuge . . . . .	299
7.11.6 Quellen systemd-unit-Datei . . . . .	302
7.12 systemd-service . . . . .	304
7.12.1 service-Unit anlegen . . . . .	304
7.12.2 Sektion Service . . . . .	304
7.12.3 Quellen systemd-service . . . . .	308
7.13 systemd-mount . . . . .	309
7.13.1 Inhalt der mount-Unit . . . . .	310
7.13.2 Inhalt der automount-Unit . . . . .	311
7.13.3 Beispiele . . . . .	312
7.13.4 Quellen systemd-mount . . . . .	315
7.14 systemd-target - Ziel-Unit . . . . .	317
7.14.1 Besonderheiten . . . . .	318
7.14.2 Quellen systemd-target . . . . .	319
7.15 systemd-path . . . . .	320
7.15.1 Benötigte Dateien . . . . .	320
7.15.2 path-Unit Optionen . . . . .	320
7.15.3 path-Unit anlegen . . . . .	322
7.15.4 service-Unit für path . . . . .	323
7.15.5 path-Unit eingliedern . . . . .	325
7.15.6 service-Unit manuell ausführen . . . . .	326
7.15.7 Quellen systemd-path . . . . .	327
7.16 systemd-timer . . . . .	328
7.16.1 Benötigte Dateien . . . . .	328
7.16.2 service-Unit für timer . . . . .	328
7.16.3 timer-Unit anlegen . . . . .	329
7.16.4 timer-Unit als cron Ersatz . . . . .	331
7.16.5 Quellen systemd-timer . . . . .	331
7.17 systemd-boot . . . . .	332
7.17.1 systemd-boot installieren . . . . .	335
7.17.2 Vorbereitung des Systems . . . . .	335
7.17.3 Konfiguration . . . . .	339
7.17.4 GRUB entfernen . . . . .	341

7.17.5 systemd-boot und Btrfs . . . . .	343
7.17.6 Weitere Informationen . . . . .	343
7.18 Systemjournal . . . . .	344
7.18.1 journald . . . . .	344
7.18.2 journald über das Netzwerk . . . . .	345
7.18.3 journald.conf . . . . .	345
7.18.4 journalctl . . . . .	346
7.18.5 journalctl beherrschen . . . . .	350
7.18.6 Quellen journald . . . . .	354

# 1 Willkommen

## 1.1 Das siduction GNU-Linux Betriebssystem

Der Name **siduction**<sup>™</sup> 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”.

Mit den ISOs wird das zu dem jeweiligen Zeitpunkt aktuelle Handbuch ausgeliefert. Da fortlaufend Änderungen in das Handbuch einfließen, lohnt sich gelegentlich ein Blick in die [Onlineversion](#).

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, das auf allen ISOs und nach der Installation im System verfügbar ist, verwendet für die langen Befehle den Zeilenumbruch.

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 bes-

ten 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 © 2006-2023 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: 2023-08-15

## 1.2 Release Notes 2024.1.0 Shine on ...

Das siduction-Team freut sich, euch ein neues Release vorzustellen. Seit dem letzten Release ist einige Zeit vergangen. Der Grund dafür ist, dass wir auf KDE Plasma 6 in Debian Unstable gewartet haben. Plasma 6 erschien zwar bereits Ende Februar 2024, die Freigabe in Unstable wurde aber durch die [t64-Transition in Debian](#), an die ihr euch bestimmt noch erinnert, um rund vier Monate verzögert. Nun ist es aber so weit und seit rund zwei Wochen ist Plasma 6.2 in Unstable verfügbar.

User und Passwort für die Live-Session lauten siducer/live

### 1.2.1 Was erwartet euch bei siduction 2024.1.0

Zunächst ein paar Worte zum neuen Artwork von siduction 2024.1.0 mit dem Titel »Shine on...«. Das bekannte Album von Pink Floyd stand hier Pate. Und wenn ihr jetzt einen Ohrwurm habt... Gern geschehen :)

Als Grundlage dient das Wallpaper [Nexus](#) von Krystian Zajdel. Wir haben es für siduction angepasst und huldigen damit den Entwicklern von KDE, die in diesem Jahr wieder enorm viel Arbeit in das Projekt gesteckt haben und es für uns weiterhin zur besten Desktop-Umgebung machen. Wenn ihr das auch so seht, denkt bitte über eine [Spende oder Förderung von KDE](#) nach.

### 1.2.2 siduction Editionen

Die Flavours, die wir für siduction 2024.1.0 anbieten, sind KDE Plasma 6.2.4.1, LXQt 2.1.0-1, Xfce 4.20, Xorg und noX. GNOME, MATE und Cinnamon haben es wieder nicht geschafft, da es keinen Betreuer innerhalb von siduction dafür gibt. Bei Interesse meldet euch bitte. Vielleicht kommen sie eines Tages zurück oder auch nicht. Natürlich sind sie weiterhin aus dem Repository installierbar.

Die veröffentlichten Images von siduction 2024.1.0 sind ein Schnappschuss von Debian Unstable, das auch den Namen Sid trägt, vom 23.12.2024. Sie sind mit einigen nützlichen Paketen und Skripten, einem auf Calamares basierenden Installer und einer angepassten Version des Linux-Kernels 6.12.6 angereichert, während systemd bei 257.1-3 steht.

### KDE Plasma 6

Plasma 6 ist mittlerweile fast zu 100 % in Unstable und Testing angekommen und wird somit auch für Debian 13 »Trixie« verfügbar sein. Obwohl Wayland der Standard-Sitzungstyp bei Plasma 6 ist, haben wir uns für X11 als Standard entschieden, weil Calamares unter Wayland derzeit das gewünschte Tastaturlayout nicht übernimmt. Das könnte fatale Auswirkungen bei verschlüsselten Installationen haben. Natürlich könnt ihr im SDDM jederzeit zu Wayland wechseln. Als Bestandsanwender habt ihr ja vermutlich bereits auf Plasma 6 aktualisiert, jetzt steht die aktuelle Plasma-Generation auch für Neuinstallationen bereit.

Bekannte Fehler: Dolphin kann derzeit über smb:// keine Verbindung aufbauen. Als Alternative bietet sich übergangsweise sftp an. Der Fehler liegt in dem Paket kio-extras.

### **Xfce 4.20**

Gerade noch so hat es das am 15. Dezember veröffentlichte [Xfce 4.20](#) nach Unstable und somit in unser neues Release geschafft. Die neue Veröffentlichung von Xfce konzentriert sich unter anderem auf die initiale, noch experimentelle Bereitstellung von Wayland. Überdies erhielt unter anderem der Dateimanager Thunar wesentliche Verbesserungen.

Bekannte Fehler: Leider ist die Implementierung von Wayland derzeit so experimentell, dass die Sitzung erst gar nicht startet. Wir haben dementsprechend Wayland blockiert. Wenn dieser Fehler behoben wird, werden wir dies in einem Punkt-Release bald wieder freischalten.

### **LXQt 2.1**

LXQt ist der leichtgewichtige Bruder von KDE Plasma. Der seit zehn Jahren entwickelte Desktop bietet in [Version 2.1.0](#) eine nutzbare, aber ebenfalls noch als experimentell eingestufte Wayland-Sitzung.

### **1.2.3 siduction-btrfs verbessert**

Bei Verwendung des Btrfs Dateisystems mit siduction ermöglichen wir es, eure Snapshots mit dem bei SUSE entwickelten Tool [Snapper](#) zu verwalten, dem im [siduction Handbuch](#) unter *Systemadministration* → *Btrfs und Snapper* ein eigenes Kapitel gewidmet ist.

siduction-btrfs kann auf Systemen mit MBR- und GPT-Partitionstabellen verwendet werden, mit oder ohne separate /boot-Partition.

Seit der Version 0.3.0 verwendet siduction-btrfs das Snapper-Plugin-Verzeichnis.

#### *Mit dem Bootmanager GRUB*

Nach einem Rollback wird im Rollbackziel mittels chroot die Datei */boot/grub/grub.cfg* neu erstellt und anschließend wird aus dem Rollbackziel heraus GRUB neu installiert. Dadurch gelangt der User mit einem einfachen Reboot direkt in das Rollbackziel. Alle anderen Subvolumen, auch das zuvor verwendete, sind über das Untermenü *siduction snapshots* erreichbar.

Wird bei Software Installation oder Upgrade die Datei */boot/grub/grub.cfg* aktualisiert, erweitert das Skript *grub-menu-title* die Menüzeile des Standardbooteintrages um das Flavor und das Subvolumen.

#### *Mit dem Bootmanager systemd-boot*

Nach einem Rollback erstellt das Skript *rollback-sd-boot* die Booteinträge. Dabei berücksichtigt es alle im Rollbackziel enthaltenen Kernel. Der Standardbooteintrag wird auf das Standardsubvolumen gesetzt.

Wird ein Subvolumen gelöscht, für das Booteinträge bestanden, werden diese entfernt.

#### *Snapper Snapshot Beschreibung*

Im Anschluss an eine APT Aktion ändert das Skript *snapshot-description* die von Snapper angezeigte Beschreibung (apt) zu einem aussagekräftigeren Text.

### **Das Changelog der Änderungen:**

- siduction-btrfs (0.2.0) unstable; urgency=medium
  - Unterstützung für systemd-boot hinzugefügt.
  - Installationsabhängigkeiten zu grub-common und grub-btrfs entfernt. Beides ermöglicht den vollständigen Wechsel von GRUB zu systemd-boot.
  - Verbesserte Beschreibung von Schnappschüssen im Snapper.
- siduction-btrfs (0.3.0-1) unstable; urgency=medium
  - Umgeschrieben mit dem Ziel, das Snapper-Plugin-Verzeichnis zu verwenden.
  - Unterstützung für eine Boot-Partition bei Verwendung von GRUB hinzugefügt.

### 1.2.4 Non-free and Contrib

Die folgenden non-free und contrib Pakete sind standardmäßig installiert:

**Nonfree:**

- amd64-microcode – Processor microcode firmware for AMD CPUs
- firmware-amd-graphics – Binary firmware for AMD/ATI graphics chips
- firmware-atheros – Binary firmware for Atheros wireless cards
- firmware-bnx2 – Binary firmware for Broadcom NetXtremell
- firmware-bnx2x – Binary firmware for Broadcom NetXtreme II 10Gb
- firmware-brcm80211 – Binary firmware for Broadcom 802.11 wireless card
- firmware-crystalhd – Crystal HD Video Decoder (firmware)
- firmware-intelwimax – Binary firmware for Intel WiMAX Connection
- firmware-iwlwifi – Binary firmware for Intel Wireless cards
- firmware-libertas – Binary firmware for Marvell Libertas 8xxx wireless car
- firmware-linux-nonfree – Binary firmware for various drivers in the Linux kernel
- firmware-misc-nonfree – Binary firmware for various drivers in the Linux kernel
- firmware-myricom – Binary firmware for Myri-10G Ethernet adapters
- firmware-netxen – Binary firmware for QLogic Intelligent Ethernet (3000)
- firmware-qlogic – Binary firmware for QLogic HBAs
- firmware-realtek – Binary firmware for Realtek wired/wifi/BT adapters
- firmware-ti-connectivity – Binary firmware for TI Connectivity wireless network
- firmware-zd1211 – binary firmware for the zd1211rw wireless driver
- intel-microcode – Processor microcode firmware for Intel CPUs

**Contrib:**

- b43-fwcutter – utility for extracting Broadcom 43xx firmware
- firmware-b43-installer – firmware installer for the b43 driver
- firmware-b43legacy-installer – firmware installer for the b43legacy driver
- iucode-tool – Intel processor microcode

**Non-Free Inhalte entfernen**

Momentan bietet der Installer keine Möglichkeit, Pakete abzuwählen, die nicht mit den DFSG, den Debian-Richtlinien für Freie Software, übereinstimmen. Das

bedeutet, dass Pakete wie etwa unfreie Firmware standardmäßig auf dem System installiert werden. Der Befehl `vrms` wird diese Pakete für dich auflisten. Du kannst nicht erwünschte Pakete manuell deinstallieren oder sie alle entfernen, indem du vor oder nach der Installation `apt purge $(vrms -s)` eingibst. Andernfalls kann später unser Skript `remove-nonfree` dies für dich tun.

### 1.2.5 Installationshinweise und bekannte Probleme

Wenn ihr eine bestehende Home-Partition (oder eine andere Datenpartition) wiederverwenden möchtet, solltet ihr dies nach der Installation und nicht im Calamares-Installer tun.

Bei einigen Intel-Grafikprozessoren auf einigen Geräten kann es vorkommen, dass das System kurz nach dem Booten in Live eingefroren ist. Um dies zu beheben, müsst ihr den Kernel-Parameter `intel_iommu=igfx_off` setzen, bevor ihr erneut bootet.

Zuletzt bearbeitet: 2024-12-23

## 1.3 Credits

### 1.3.1 Credits für siduction 2024.1.0

**Core Team:**

Torsten Wohlfarth (towo)  
Hendrik Lehmbruch (hendrikL)  
Ferdinand Thommes (devil)  
Vinzenz Vietzke (vinzv)  
Axel Konrad (akli)

**Früher haben beigetragen:**

Alf Gaida (agaida) (eaten by the cat)  
Axel Beu 2021†  
Markus Meyer (coruja)

**Code, Ideen und Unterstützung:**

Markus Meyer (coruja)  
der\_bud  
se7en  
davydych  
tuxnix  
tobilinuxer

### 1.3.2 Credits für das Handbuch

**Core Team:**

Torsten Wohlfarth (towo)  
Hendrik Lehmbruch (hendrikL)  
Ferdinand Thommes (devil)  
Vinzenz Vietzke (vinzv)  
Axel Konrad (akli)

**Credits 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

### **1.3.3 Unser Dank gilt allen Beteiligten und unseren treuen Usern**

Wir möchten uns bei euch, bei allen Testern und all den Menschen bedanken, die uns über die Jahre hinweg unterstützt haben. Dieses Release ist auch dein Verdienst. Wir danken zudem mit diesem Release der KDE-Gemeinschaft, die eine hervorragende Desktop-Umgebung bereitstellt.

Im Namen des siduction-Teams:  
Ferdinand Thommes

Zuletzt bearbeitet: 2024-12-23

## 1.4 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:

### 1.4.1 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.

### 1.4.2 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 6697
```

- 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.
- Du hast bereits eine Identität bei **matrix**, dann benutze diesen Link.  
<https://matrix.to/###siduction-de:matrix.sp-codes.de>

### 1.4.3 Nützliche Helfer im Textmodus

Normalerweise verwendet man den Textmodus `systemctl isolate multi-user.target`, wenn man ein dist-upgrade durchführen möchte, oder gezwungenmaßen, 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 und Texteditor im Text-Modus (tty). 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.sources /etc/apt/sources✓  
    .list.d/debian.sources_$(date +%F)  
  
    anschließend  
  
$ mcedit /etc/apt/sources.list.d/debian.sources
```

Nun kann die Datei bearbeitet und gespeichert werden. Die Änderungen werden sofort wirksam.

Weitere Informationen auf der Manpage:

```
$ man mc
```

#### 1.4.4 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 6697 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 \(Deutsch\)](#)

### 1.4.5 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.

### 1.4.6 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
```

### 1.4.7 Nützliche Links

[Debian Referenzkarte - zum Ausdruck auf ein Einzelblatt](#)

[HOWTOs von der Debian-Seite](#) (ist automatisch auf Deutsch, wenn Browser lokalisiert 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: 2026-03-03

## 1.5 Skripte in siduction

siduction enthält einige Skripte, die administrative Aufgaben unterstützen und bei der Fehlersuche behilflich sind.

### 1.5.1 Vom Benutzer ausführbar

- **chroothelper**

*Wechsel in eine chroot-Umgebung*

*chroothelper* vereinfacht den Wechsel in eine chroot-Umgebung auf dem Live-System. Es durchsucht die Festplatten nach installierten Linux Betriebssystemen und bietet einen Wechsel in diese mittels chroot an.

- **sshactivate** und **sshdeactivate**

*Startet und aktiviert sshd oder stoppt und deaktiviert sshd*

*sshactivate* erzeugt sshd-Systemschlüssel, startet und aktiviert den ssh-Daemon und fragt nach dem Benutzerpasswort, um den Fernzugriff auf das (Live-)System zu ermöglichen.

*sshdeactivate* stoppt und deaktiviert den ssh-Daemon.

- **fw-detect**

*Prüft den Firmwarestatus im System.*

*fw-detect* scannt die geladenen Module auf Ihrem System, prüft, ob sie Firmware benötigen und ob diese im richtigen Verzeichnis vorhanden ist. Standardmäßig gibt *fw-detect* nur Informationen über Module aus, die allem Anschein nach Firmware benötigen.

- **siduction-irc**

*Startet eine IRC-Sitzung zu #siduction-de*

*siduction-irc* startet eine IRC-Sitzung mit folgenden Clients (je nach Umgebung): HEXCHAT, IRSSI, KONVERSATION, KVIRC, WEECHAT, XCHAT und verbindet sich mit unserem IRC-Kanal #siduction-de auf [OFTC](#).

- **remove-nonfree**

*Für eine DFSG konforme Installation*

*remove-nonfree* macht die Installation DFSG-konform. Das ausführen, kann zu einem nicht funktionierenden WLAN oder zu Problemen mit der Grafikkarte führen.

- **kernel-remover**

*Entfernt siduction Kernel*

*kernel-remover* entfernt unbenutzte Kernel aus dem laufenden System. Es bietet eine Auswahl der Kernel, entfernt die gewählten und aktualisiert das Bootmenü.

- **onoff**

*Ersatz für den entfernten Befehl init.*

*onoff* stellt die durch systemd 258~rc1-1 entfernte Funktionalität von init 0 bis init 6 für den Benutzer in einem laufenden System wieder zur Verfügung.

### 1.5.2 Im Hintergrund laufend

- **siduction-btrfs**

*Überprüft und aktualisiert das Bootmenü*

*siduction-btrfs* ist eine Zusammenstellung von Skripten und systemd Units, die auf einer Installation in das Btrfs Dateisystem aktiv wird. Nach einem Snapshot oder Rollback wird das Bootmenü überprüft und gegebenenfalls aktualisiert. *siduction-btrfs* arbeitet im Hintergrund ohne Benutzereingaben. Ab Version 0.2.0-5 unterstützt *siduction-btrfs* auch den Bootmanager *systemd-boot* und generiert Booteinträge nach einem Rollback für das Rollbackziel. Mehr Informationen finden sich in unserer [Handbuchseite zu systemd-boot](#).

Zuletzt bearbeitet: 2026-03-01

## 2 Quickstart

Für schnell entschlossene User hier eine stichpunktartige Zusammenfassung wichtiger Informationen zur Verwendung des Live Mediums und Installation von siduction.

- [Essenzielle Kapitel](#)
- [siduction-Kernel und Softwarepakete](#)
- [Konfiguration von Netzwerken](#)
- [Installation kurz und schnell](#)

Zuletzt bearbeitet: 2023-11-13

## 2.1 Einleitung

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, Hilfsprogramme und Dokumentationen.

Die Distribution siduction richtet sich an Benutzer, die bereits einige Erfahrung mit Linux gesammelt haben und sich nicht vor dem Einsatz der Kommandozeile scheuen.

### 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.

#### 2.1.1 Essenzielle Kapitel

Für Nutzer, die neu bei siduction sind, empfehlen wir dringend die Lektüre der folgenden Kapitel. Sie enthalten neben grundlegenden Informationen zu Linux wichtige, siduction spezifische Einstellungen und Vorgehensweisen.

- [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 Life-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 Firmware installiert werden kann.
- [Internetverbindung](#) - Beschreibt, wie man sich mit dem Internet verbinden kann.
- [Paketmanager und Systemaktualisierung](#) - Beschreibt, wie neue Software installiert und das System aktualisiert werden kann.

Zuletzt bearbeitet: 2023-11-13

## 2.2 Kernel und Softwarepakete

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/>. siduction basiert auf Debian Sid, was dazu führen kann, dass in seltenen Fällen Debian-Pakete Fehler enthalten. Deshalb bieten wir gegebenenfalls Pakete an, die temporär fehlerhafte Debian-Pakete ersetzen.

### 2.2.1 Die Verwaltung von Softwarepaketen

siduction richtet sich nach den Debian-Regeln bezüglich der Paketstruktur 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:

[Programmpakete suchen](#) .

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 aktualisiert bzw. mit 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.

### 2.2.2 Aktualisierung des Systems - upgrade

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

```
systemctl isolate multi-user.target
```

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:

```
systemctl isolate graphical.target
```

“*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](#).

Zuletzt bearbeitet: 2025-09-19

## 2.3 Konfiguration von Netzwerken

Der in allen graphischen Oberflächen von siduction integrierte **NetworkManager** bietet eine schnelle Konfiguration von Netzwerkkarten (WLAN und Ethernet). Im Terminal bietet das Skript **nmcli** Zugang zur Funktionalität des NetworkManager.

Drahtlose Netzwerke werden gescannt. Man kann sich mit den gefundenen Netzwerken verbinden und Einstellungen zur Verschlüsselungsmethode, zum Internet Protokoll IPv4 oder IPv6 und zu einem Proxy Server vornehmen. Als Backend kommt **iwid** zum Einsatz. 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 gegeben.

Auf dem Live Medium und nach der Installation ist der NetworkManager mit seinem Backend **iwid** bereits konfiguriert und einsatzbereit.

### 2.3.1 NetworkManager

In der graphischen Oberfläche befindet sich der NetworkManager in der Taskleiste. Er ist größtenteils selbsterklärend.

Mit **nmcli** oder **nmtui** steht ein leistungsfähiger Kommandozeilenclient für den täglichen Gebrauch des NetworkManagers zur Verfügung. **nmtui** bietet eine benutzerfreundliche ncurses Oberfläche innerhalb des Terminals. Auch hier ist die Funktion größtenteils selbsterklärend. Falls das Skript nicht vorhanden ist, installiert man es mit:

```
apt install network-manager
```

Der Startbefehl in der Konsole ist **nmcli**.

Eine detaillierte Beschreibung zur Bedienung findet sich unter [Netzwerk - nmcli](#).

Hat man alle notwendigen Informationen zur Hand, genügt auch eine einzige Kommandozeile.

Beispiel:

Device Name = wlan0

Device Type = wifi

SSID = HOME\_WLAN

Passwort = s3Hrg3he!m

Befehl zum Aufbau der WLAN Verbindung:

```
nmcli device wifi connect "HOME_WLAN" password "s3Hrg3he!m"
```

Die Verbindung trennt man mit:

```
nmcli device down wlan0
```

NetworkManager speichert die einmal eingegebenen Verbindungsdaten. Befindet man sich erneut in Reichweite dieses Routers, stellt NetworkManager die Verbindung automatisch wieder her. Um das Verhalten zu ändern bitte die Manpages

**man NetworkManager**

und

**man nm-settings**

konsultieren.

Zuletzt bearbeitet: 2023-11-13

## 2.4 Schnellinstallation

### 2.4.1 Fünf Schritte zum Ziel

- 1) Auf der [siduction Downloadseite](#) das bevorzugte Flavour aussuchen und die ISO-Datei, sowie die Datei mit der zugehörigen Checksumme herunterladen.

Anschließend den Download wie [hier beschrieben](#) prüfen.

- 2) Die ISO-Datei auf ein bootbares Medium übertragen.  
Hilfe bieten die Anleitungen für [einen USB Stick oder eine SD Karte](#) oder zum [Brennen auf eine DVD](#).

- 3) Das so erstellte siduction Live Medium booten.

**Wichtig:**

Während des Bootvorgangs das EFI Menü aufrufen (je nach Hersteller eine der Tasten **Entf**, **F2**, **F11**... drücken).

Im EFI Menü sowohl den *CSM (Compatibility Support Module)* Modus, als auch *Secure Boot* deaktivieren.

Ebenfalls im EFI Menü das Live Medium zum booten auswählen.

- 4) Im Bootmenü des Live Mediums die gewünschte Sprache einstellen und siduction booten.

Der Benutzer im Live Medium ist **siducer** sein Passwort lautet **live**.

Für den Benutzer **root** (Systemadministrator) ist im Live Medium kein Passwort gesetzt. Im Terminal entweder **sudo <Befehl>** ausführen, oder mit der Eingabe von **su** zu root werden.

- 5) Ein Doppelklick auf das Icon **System installieren** startet die Installation auf die Festplatte.

Wenn manuell partitioniert wird, darauf achten, dass die erste Partition die EFI System Partition (ESP) und die Markierung *bootfähig* gesetzt ist. Sie muss unter **/boot/efi/** eingehangen werden.

Zuletzt bearbeitet: 2025-03-25

## 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 des Live-Mediums](#),  
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 Spiegelservers 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: 2023-11-09

## 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 und Intel CPUs

(achte auf das "lm"-Flag in /proc/cpuinfo oder nutze inxi -v3).

#### Speicheranforderungen

KDE Plasma	mindestens	4 GByte	RAM
Xfce	mindestens	4 GByte	RAM
LXQt	mindestens	512 MByte	RAM
Xorg	mindestens	512 MByte	RAM
NoX	mindestens	256 MByte	RAM

Mindestens 5 GByte Festplattenspeicher für NOX.

Mindestens 15 GByte Festplattenspeicher für alle anderen Varianten.

Mindestens 50 GByte Festplattenspeicher bei Installation auf einer mit Btrfs formatierten Partition.

### Sonstiges

VGA Grafikkarte mit mindestens 640x480 Pixel Auflösung und ein 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 Network-Manager, Connman oder iwd zur Verfügung. Als WLAN-Daemon kommt [iwd](#) zum Einsatz.

Zur Partitionierung von Festplatten werden fdisk und [cfdisk](#) sowie parted 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: 2026-02-17

## 3.2 Live-Medium 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 Chroot Helfer

Ein sehr hilfreiches Tool auf dem Live-Medium ist der *Chroot Helfer*. Wer kann schon aus dem Stegreif die Befehle zum Starten einer chroot ausführen, die meisten von uns nicht. Wenn wir also das nächste Mal unser System reparieren müssen, benutzen wir das Icon *Chroot Helfer*. Es öffnet ein Terminal Fenster und wir werden aufgefordert eine Partition auszuwählen. Nach einer Sicherheitswarnung leitet uns der *Chroot Helfer* in die entsprechende Installation um.

### 3.2.3 Mit root-Rechten auf dem Live-Medium

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:~# gparted &
```

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](#)

### 3.2.4 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.

### 3.2.5 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-gewünschte-paket>
```

oder ein vorangestelltes **sudo** vor die Befehle.

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

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

Zuletzt bearbeitet: 2024-08-25

## 3.3 Bootoptionen Cheatcodes

### Info

Diese Handbuchseite enthält Tabellen zu den Bootoptionen für

1. siduction spezifische Parameter (nur Live-Medium)
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 das Live-Medium.

Bootoption	Wert	Beschreibung
blacklist	Name des Moduls	temporäre Deaktivierung von Modulen, bevor udev aktiv wird
desktop	kde, gnome, fluxbox	Desktopumgebung auswählen
fromiso		<a href="#">bitte lies "Booten 'fromiso'"</a>
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 <b>lang=ll_cc</b> oder <b>lang=ll-cc</b> bedeutet <b>ll</b> die Sprachauswahl und <b>cc</b> Tastaturlayout, Spiegelserver und Zeitzonewahl (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 des Live-Mediums
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 des Live-Mediums
nomodeset	radeon.modeset=0	ermöglicht zusammen mit <b>xmodule=vesa</b> ein sauberes Booten nach X bei Radeonkarten im Live-Mode

---

Bootoption	Wert	Beschreibung
nonetwork		verhindert die automatische Konfiguration von Netzwerkschnittstellen beim Booten
noswap		Keine Aktivierung der Swap-Partition
smouse		sucht mittels hwinfo nach seriellen Mauseingabegeräten
tz	tz=Europe/Dublin	setzt die Zeitzone. Falls die Bios- bzw. Hardwareuhr auf UTC eingestellt ist, wird <b>utc=yes</b> angegeben. Eine Liste aller unterstützter Zeitzonen kann eingesehen werden, wenn per copy & paste: <b>file:///usr/share/zoneinfo/</b> in den Browser eingegeben wird .
toram		kopiert das Live-Medium 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 $1,147 \times 1920 / 24$ dpi=92 oder für einen 15"-Bildschirm mit der Auflösung 1600x1200 ergibt das mittels $1,25 \times 1600 / 15$ 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 <a href="#">hier</a>
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 <a href="#">Runlevel - Ziel-Unit</a>
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: 2023-11-10

## 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.sources` 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  
SHA256SUM  
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.sources** enthält die Downloadlinks zu den Quellcodedateien der verwendeten Pakete.

Download-Links und Spiegelserver findet man auch auf [siduction.org](http://siduction.org)

Das Tar-Archiv mit den Quellen ist für den interessant, der siduction weiterverbreiten 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:6697 #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: 2025-02-12

## 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.

### 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 `journalctl`. 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 ...  
[...]  
kernel: scsi 1:0:0:0: Direct-Access   Intenso   Alu Line ...  
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 ...
```

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 bs=1M oflag=sync status=progress 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 abzuliegen.

Wir benutzen als root den Befehl `parted` um eine neue MSDOS Partitionstabelle zu erstellen.

**Achtung**

Dabei gehen alle Daten auf diesem Medium verloren.

```
parted /dev/sdb mktable msdos
```

Nach Bestätigung der Sicherheitsabfrage wird die neue Partitionstabelle geschrieben. Der nächste Befehl `cfdisk /dev/sdb` startet das Programm cfdisk, mit dem wir entsprechend der folgenden Daten die gewünschten Partitionen anlegen. (Siehe die Handbuchseite [Partitionieren mit cfdisk](#))

1. Partition:

Startsektor: 2048 (Voreinstellung)

Größe: 3G (3 GB, etwas größer als die ISO-Abbilddatei)

primary

Partition Type: W95 FAT32 (LBA) (c)

bootable

2. Partition:

Startsektor: xxxxxxxx (Voreinstellung, 1. Sektor nach der vorherigen Parti-

tion)

Größe: xxxxxxxx (Voreinstellung, die maximal mögliche Größe)

Partition Type: Linux (83)

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: 2026-02-17

## 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, 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 zu meist 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: 2023-11-10

## 3.7 Life-DVD ohne GUI brennen

### WICHTIGE INFORMATION:

siduction, als Linux-LIVE-DVD, ist sehr stark komprimiert. Aus diesem Grund muss besonders auf die Brennmethode des ISO-Abbilds geachtet werden. Wir empfehlen hochwertige Medien, 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 [...] drive...
Detected CD-R drive: /dev/sr0
[...]
Vendor_info      : 'HL-DT-ST'
Identification  : 'DVD-RAM 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 `cdrecord`:

```
$ cdrecord scanbus
Cdrecord version 1.2.4 - (C) Andreas Mueller
/dev/sr0 : HL-DT-ST, DVD-RAM 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 "`/dev/scd1`") 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 filename.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 datei8 datei9
```

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: 2023-11-10

## 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 fdisk](#) nach dem UEFI-GPT Standard oder mit den herkömmlichen BIOS mit MSDOS-Partitionstabellen.
- [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: 2026-02-22

## 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 Festplattenspeicher und 4 GB Arbeitsspeicher ist man zur Zeit auf der sicheren Seite. Bei Installation auf einer mit Btrfs formatierten Partition raten wir zu 50 GByte Festplattenspeicher.

Auf PCs mit maximal 2 GB RAM sollte eine Swap-Partition angelegt werden. Mehr als 4 GB Swap wird normal nicht benötigt und ist nur bei Suspend-to-Disk und Serversystemen wirklich sinnvoll.

### 4.1.3 Partitionierung und Dateisysteme

Die Handbuchseite [Partitionierung von Installationsmedien](#) enthält einige Beispiele die verschiedene Größen der Festplatten berücksichtigen und Anregungen zur Aufteilung bei Single- und Dualbootssystemen geben. Im Kapitel [Dateisysteme der Partitionen](#) beschreiben wir, welche Dateisysteme auf den Partitionen in der jeweiligen Situation sinnvoll sind.

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  
[cfdisk](#), ein Programm für das Terminal mit einer benutzerfreundlichen ncurses Oberfläche. Geeignet für UEFI Hardware mit GPT Partitionstabellen und für ältere Hardware oder kleinere Laufwerke wie USB Sticks und Speicherkarten mit BIOS und MSDOS Partitionstabellen.

### 4.1.4 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  
e $(uname -r) >/home/username/installed.txt
```

Am besten wird diese Textdatei auf einen USB-Stick oder einen Datenträger nach Wahl kopiert.

Auf der Ziellinstallation 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.

Möchte man eine alte siduction Installation durch eine neue ersetzen, führt die Verwendung der `installed.txt` des alten Systems auf dem neuen mit großer Wahrscheinlichkeit zu schwerwiegenden Problemen. Besser ist es eine `installed-old.txt` von dem alten, und eine `installed-new.txt` von dem neuen System mittels *diff* zu vergleichen. Anschließend installiert man die gewünschten Pakete auf der neuen siduction Installation nach.

#### 4.1.5 Das Installationsprogramm Calamares

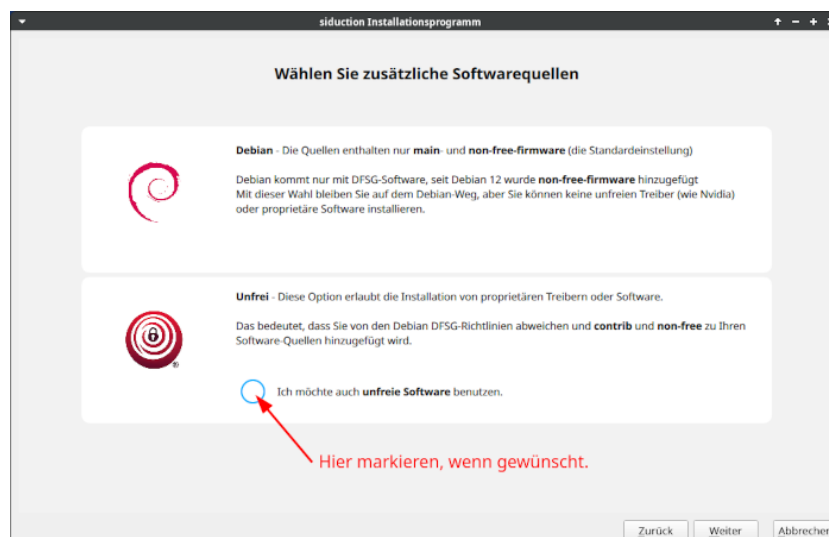
Während der Installation sollte, wenn möglich, der Computer mit dem Internet verbunden sein, weil Calamares den GeoIP Service verwendet um Voreinstellungen für die Lokalisation und Zeit zu ermitteln.

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.

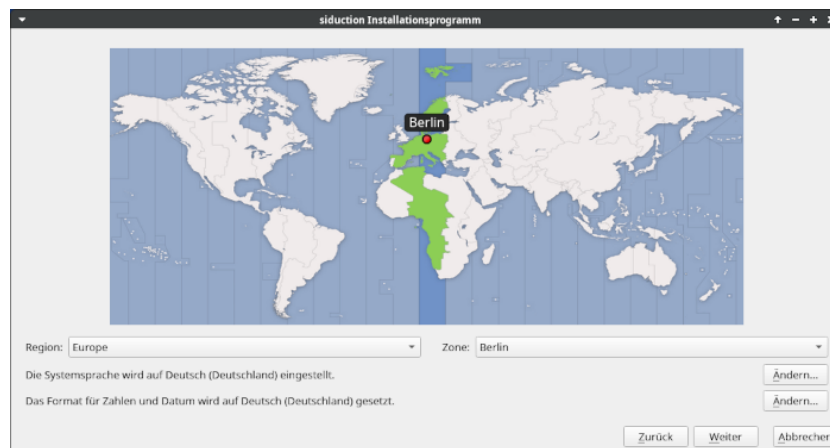


Sofern eine Internetverbindung besteht, sollte hier bereits die richtige Sprache eingestellt sein.

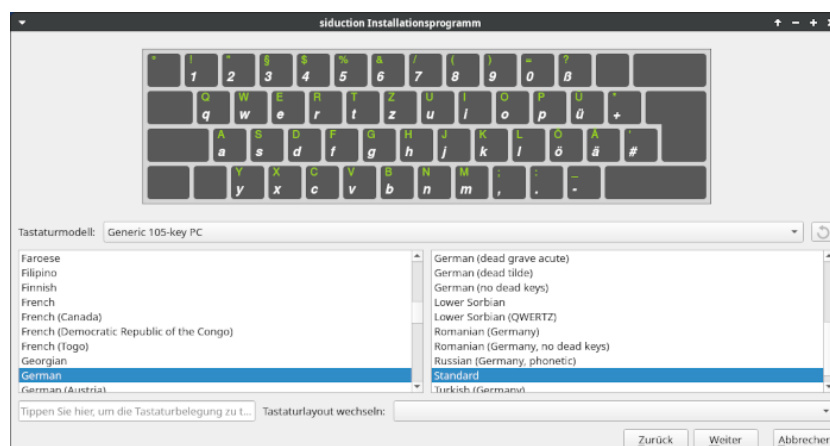
3. Als nächstes besteht die Möglichkeit zusätzliche, unfreie Software Quellen zu wählen. Bei Aktivierung dieser Option werden die Quellen *contrib* und *non-free* auch aktiviert und es ist möglich unfreie Treiber (z. B. Nvidia) und proprietäre Software zu installieren.



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



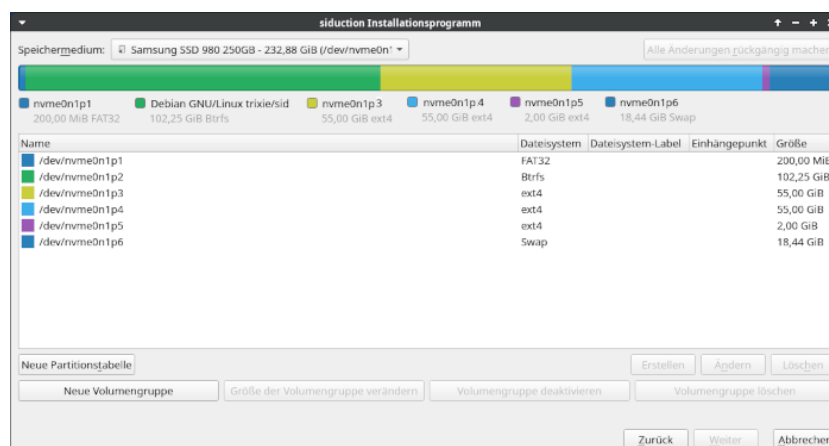
5. Es folgen die Einstellungen zur Tastatur. Im oberen Teil wird die Tastatur graphisch dargestellt und die Änderungen werden sofort sichtbar. Ganz unten befindet sich eine Eingabezeile um das Tastaturlayout zu testen.



6. 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

**nvme0n1p1** für **/boot/efi**

**nvme0n1p4** für **/** (root)

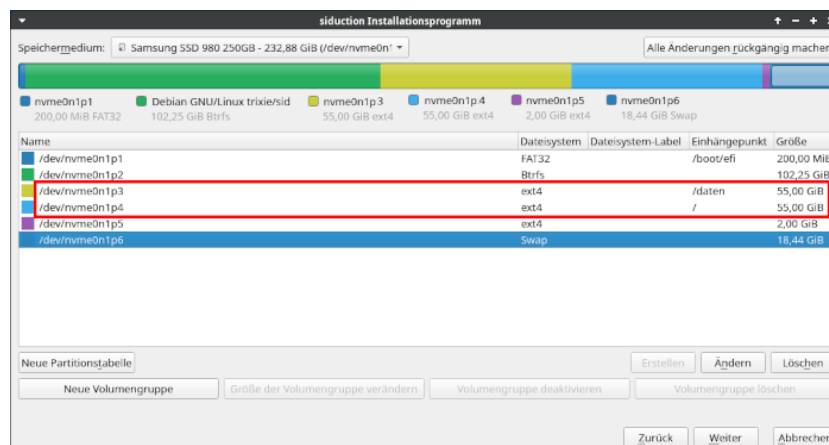
**nvme0n1p3** für **/daten** gemeinsam mit dem bereits auf **nvme0n1p2** vorhandenem Linux

Nach Auswählen der betreffenden Partition und Betätigen des Schalters **Ändern** öffnet sich ein Fenster, in dem wir den oben bezeichneten Mountpoint eintragen und für **nvme0n1p4** auch die Formatierung mit dem Dateisystem **ext4** vornehmen. Die Partition **nvme0n1p3** wird nicht formatiert, da

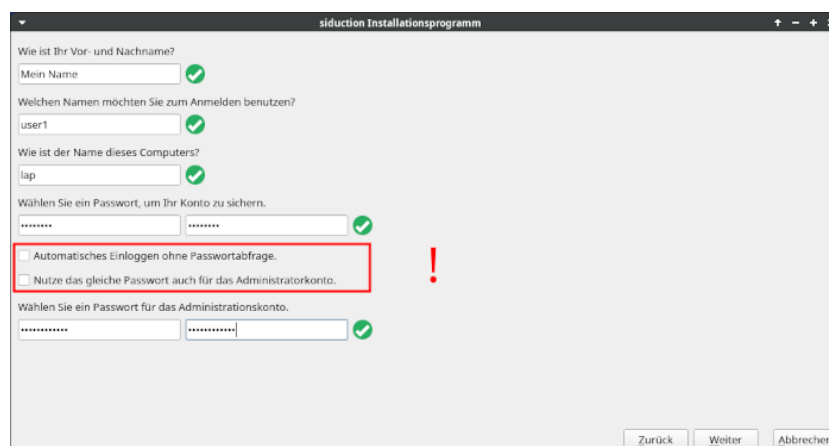
wir die dort schon abgelegten Daten gemeinsam mit dem bereits vorhandenen Linux nutzen möchten.

Die Swap-Partition `nvme0n1p6` brauchen wir nicht bearbeiten, da sie während der Installation automatisch erkannt und integriert wird.

Das Ergebnis unserer Bemühungen sehen wir im nächsten Bild.



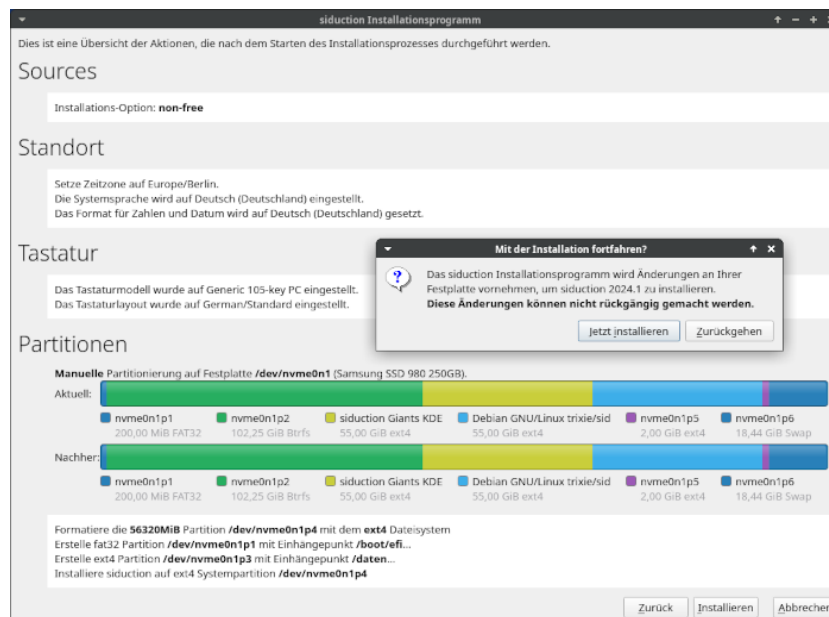
- Als nächstes werden Benutzername, Anmeldenamen, Computernamen, Benutzerpasswort und Root-Passwort festgelegt (bitte gut merken!). Die Passwörter sollen aus Sicherheitsgründen nicht zu einfach gewählt werden. Weitere Benutzer können nach der Installation in einem Terminal mit `adduser` hinzugefügt werden.



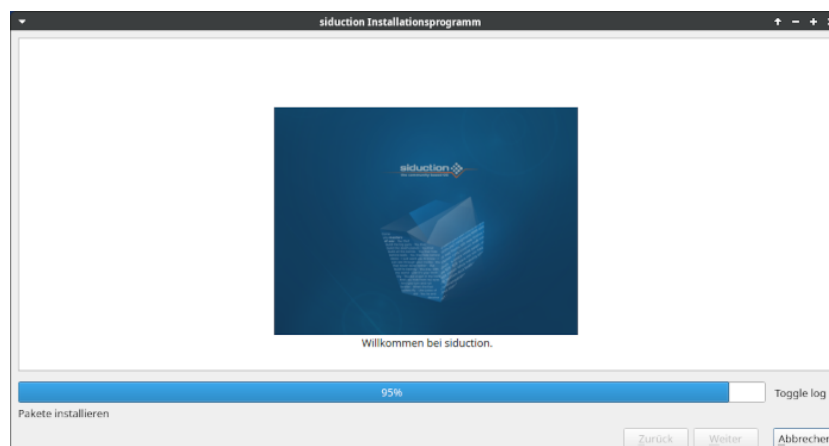
Vor der Verwendung der beiden Optionen  
*“Automatisches Einloggen ohne Passwortabfrage”* und  
*“Nutze das gleiche Passwort auch für das Administratorkonto”*

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!

- 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.



- 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.



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



Vor dem Reboot den USB Stick mit dem Live Medium entfernen!

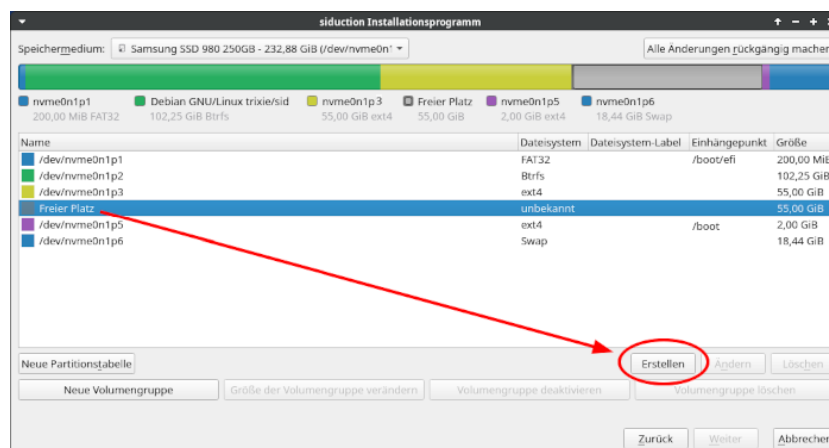
#### 4.1.6 System verschlüsseln

Die oben unter 6 beschriebene Partitionierung gestaltet sich nun geringfügig anders.

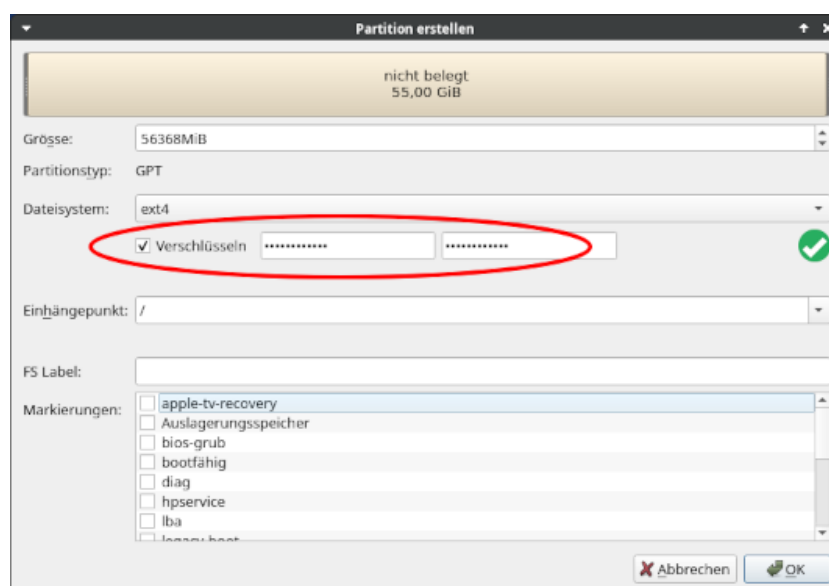
Wir benutzen auch hier die Option "*Manuelle Partitionierung*". Das verschlüsselte System erfordert mindestens drei Partitionen. Entsprechend der oben verwendeten Partitionierung sind dies:

`/dev/nvme0n1p1` unverschlüsselt und eingehangen unter `/boot/efi`,  
`/dev/nvme0n1p5` unverschlüsselt und eingehangen unter `/boot`,  
`/dev/nvme0n1p4` für das verschlüsselte System.

Die Partition `/dev/nvme0n1p4` benötigt ein anderes als das bisher verwendete Dateisystem. Deshalb löschen wir im ersten Schritt die Partition und erstellen in dem leeren, unbenutzten Bereich eine 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 7 beschrieben, fort.

#### 4.1.7 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 <nutzernamen>
```

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

Zuletzt bearbeitet: 2026-02-22

## 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 installiert 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.

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

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

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.

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

Davon abweichend empfehlen wir mindestens 20 GB Speicherplatz bei einer Installation in das Dateisystem **Btrfs** und der Verwendung von `snapper`. 50 GB

oder mehr, sind sinnvoll, wenn Sie siduction auf **Btrfs** über einen längeren Zeitraum nutzen wollen und viele Schnappschüsse gehalten werden.

### 4.3.2 Beispiele mit verschiedenen Plattengrößen

Es gibt sehr viele gute Möglichkeiten seine Platten aufzuteilen. Diese Beispiele sollten einen ersten Einblick bieten. Sie beziehen sich auf Partitionstabellen vom Typ "GPT". Die erste Partition auf der ersten Festplatte ist für den Bootvorgang zwingend notwendig.

#### Laptop mit 8 GB RAM, Linux allein

##### 250 GB Festplatte:

Partition	Größe	Dateisystem	Verwendung
1	300 MB	FAT32	EFI-System (ESP)
2	40 GB	ext4	/
3	200 GB	ext4	Daten
4	10 GB	Linux Swap	Linux Swap

#### Desktop PC, Linux allein

##### 500 GB Festplatte:

Partition	Größe	Dateisystem	Verwendung
1	300 MB	FAT32	EFI-System (ESP)
2	40 GB	ext4	/
3	456 GB	ext4	Daten
4	4 GB	Linux Swap	Linux Swap

#### Desktop PC, Linux allein

##### 500 GB Festplatte mit Btrfs-Snapshot:

Partition	Größe	Dateisystem	Verwendung
1	300 MB	FAT32	EFI-System (ESP)
2	496 GB	Btrfs	/
3	4 GB	Linux Swap	Linux Swap

**Desktop PC, Linux allein****500 GB Festplatte mit Btrfs-Snapshot und systemd-boot:**

Partition	Größe	Dateisystem	Verwendung
1	300 MB	FAT32	EFI-System (ESP)
2	495 GB	Btrfs	/
3	1 GB	ext4	XBOOTLDR
4	4 GB	Linux Swap	Linux Swap

Falls ein Dual-Boot mit MS Windows™ angelegt wird, muss MS Windows immer als erstes System auf die Festplatte installiert werden. Die ersten vier Partitionen unserer Beispiele sollen direkt aufeinander folgend am Anfang der Festplatte liegen. Danach folgen die Partitionen für Linux und gemeinsam genutzte Daten.

Siehe auch [Microsoft: UEFI/GPT Partitionierung, Windows 11](#).

**Desktop PC, Dual-Boot (MS Windows und Linux)****1 TB Festplatte:**

Partition	Größe	Dateisystem	Verwendung
1	300 MB	FAT32	EFI-System (ESP)
2	16 MB	ohne	Windows MSR
3	50 GB	NTFS	Windows System
4	1 GB	NTFS	Windows Recovery
5	415 GB	exFAT	Daten für Windows und Linux
6	30 GB	ext4	/ (Linux System)
7	500 GB	ext4	Daten für Linux
8	4 GB	Linux Swap	Linux Swap

**Laptop mit 32 GB RAM, Dual-Boot mit MS Windows und Linux****1 TB Festplatte:**

Partition	Size	Filesystem	Verwendung
1	300 MB	FAT32	EFI-System (ESP)
2	16 MB	ohne	Windows MSR
3	50 GB	NTFS	Windows System

Partition	Size	Filesystem	Verwendung
4	1 GB	NTFS	Windows Recovery
5	499 GB	exFAT	Daten für MS Windows und Linux
6	30 GB	ext4	/ (Linux System)
7	380 GB	ext4	Daten für Linux
8	40 GB	Linux Swap	Linux Swap

### 4.3.3 Dateisysteme der Partitionen

Als Partitionstabelle sollte der Typ *“GPT”* gewählt werden. So kann man die Vorteile gegenüber *“MBR”* nutzen. Nur bei alter Hardware ist *“MBR”* noch sinnvoll.

#### Linux Swap

Eine **swap**-Partition entspricht in der Funktionalität etwa der Auslagerungsdatei bei Windows, ist aber weit effektiver als diese. Ihre Größe richtet sich nach dem installierten System und den Anforderungen des Benutzers. Einige Beispiele:

- Für Notebooks, die per **hibernate** in den Ruhezustand versetzt werden sollen, empfehlen wir eine swap Partition die wenigstens ein GByte oder 25% größer ist als das RAM.
- Aktuelle Desktop PCs, die *nicht* in den Ruhezustand versetzt werden sollen und über ausreichend RAM verfügen (je nach Verwendung ab 16 GByte), benötigen keine swap Partition mehr.
- Für Desktop PCs mit sehr wenig RAM gilt nach wie vor die Faustregel, dass die swap Partition zweimal so groß sein sollte wie das verwendete RAM.

#### ext4

Das **ext4** Dateisystem ist das Default-Dateisystem bei siduction. Dies gilt für alle Partitionen, wenn ausschließlich Linux Betriebssysteme verwendet werden.

#### Btrfs

**Btrfs** kann an Stelle von **ext4** verwendet werden. Zusammen mit dem Programm **Snapper** bietet es die Möglichkeit Snapshots des Dateisystems zu erstellen, die

anschließend im Bootmanager Grub auswählbar sind. Man benötigt eine ausreichend große Festplatte. Siehe auch [Systemadministration Btrfs](#).

### NTFS

Für eine Windows-Installation müssen die dafür vorgesehenen Partitionen mit *NTFS* formatiert werden. Siduction kann lesend und schreibend auf die Daten zugreifen. Für Windows ist es das Standarddateisystem.

### exFAT

Ein Dateisystem, das von Microsoft entwickelt wurde, und in vielen Arten von Speichergeräten wie SD-Karten und USB-Flash-Laufwerken verwendet wird. Im Jahr 2019 wurden die Patente daran freigegeben und in der Folge unterstützt Linux ab Kernel 5.4 *exFAT*. Es eignet sich auch sehr gut für Partitionen, auf die unterschiedliche Betriebssysteme zugreifen sollen.

### HFS+

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

#### 4.3.4 Partitionierungsprogramme

##### Achtung

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

### 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 Desktop, einfach zu bedienen und genauso umfangreich wie Gparted.

### **fdisk / cfdisk**

Ein Konsolenprogramm für Partitionstabellen vom Typ *“msdos-MBR”* und *“GPT-UEFI”*.

fdisk ist das klassische Textmodus-Programm. cfdisk hat eine benutzerfreundlichere ncurses-Oberfläche.

Hinweis: Partitionstabellen vom Typ *“msdos-MBR”* sollten nur noch für alte Hardware, die *“GPT-UEFI”* nicht unterstützt, oder für kleinere Laufwerke wie USB Sticks und Speicherkarten verwendet werden. Die Handbuchseite [Partitionieren mit Cfdisk](#) liefert weitere Informationen zum Programm.

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
```

#### **4.3.5 Weiterführende Infos**

[Die umfassende englischsprachige Dokumentation von GParted](#)

[Microsoft: UEFI/GPT Partitionierung, Windows 11](#)

Für weitere Partitionierungsoptionen siehe:

- Logical Volume Manager [LVM-Partitionierung](#)
- [Partitionieren mit cfdisk](#) zur Unterstützung von UEFI

Zuletzt bearbeitet: 2026-02-22

## 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 Partitio-

nierung, 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**

```
swapon -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>`
- **exFAT** `exfatlabel /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`.

```
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-  
-811f-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-41c1-80ce-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**, **kwrit** oder **gedit**) mit Rootrechten; in diesem Beispiel sähe der Eintrag so aus:

```
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
```

Anschließend informieren wir den Kernel über die Änderungen an der **/etc/fstab** mit dem Befehl **systemctl daemon-reload**.

#### 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: 2023-11-07

## 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](#).

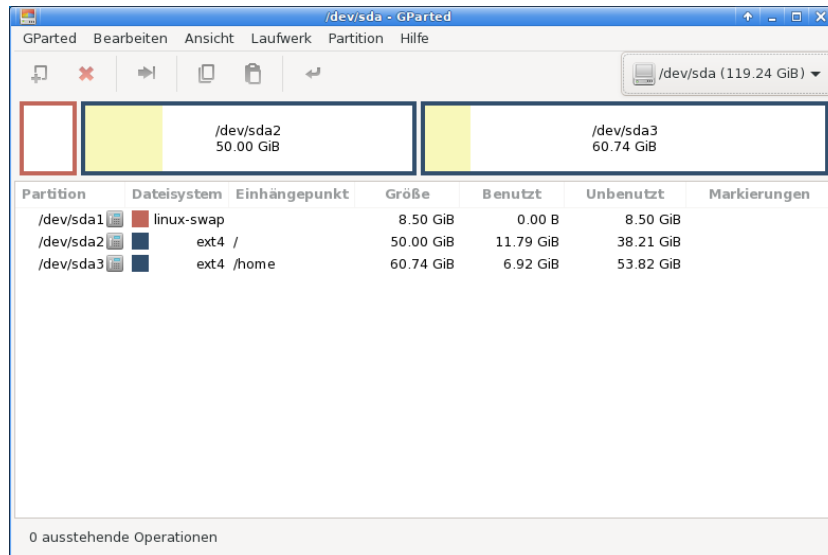
### 4.6.2 GParted verwenden

Der Programmstarter für GParted befindet sich in

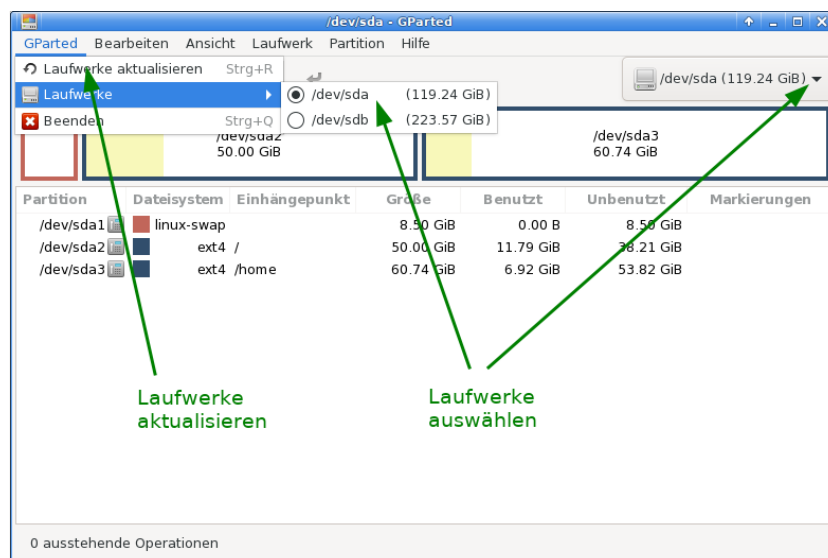
- **KDE, LXQt, XFCE**  
im Anwendungsmenü - System - GParted
- **Gnome**  
in Anwendungen - Gparted

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.



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

Dieser Menüpunkt bietet die Anzeigeeoptionen **Laufwerksinformationen** und **Anstehende Operationen**.

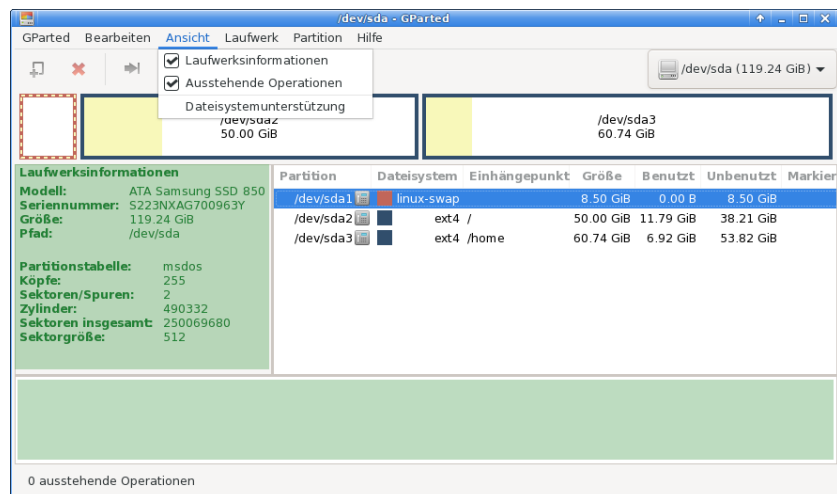
- Laufwerksinformationen (“Device Information”)

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.

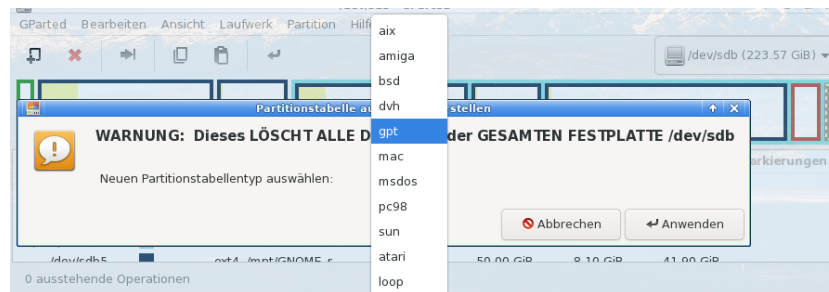


#### • Laufwerk

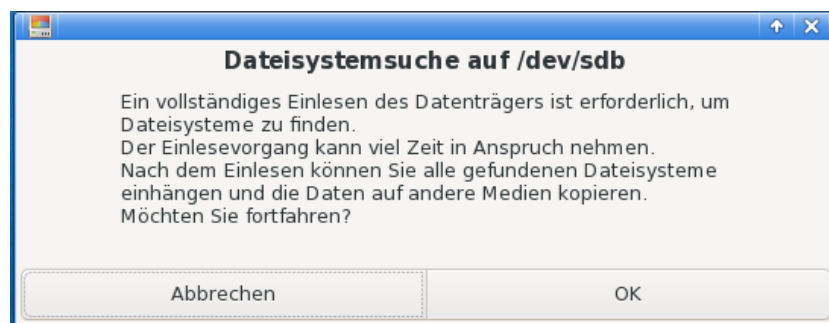
Hinter dem Menüpunkt **Partitionstabelle erstellen** verbergen sich eigentlich zwei Optionen

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.



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



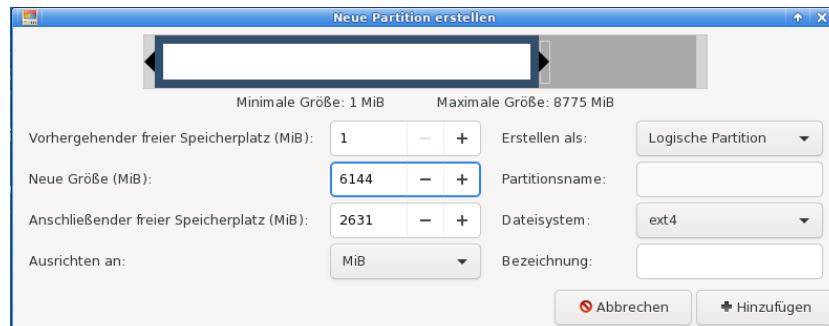
- **Partition**

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



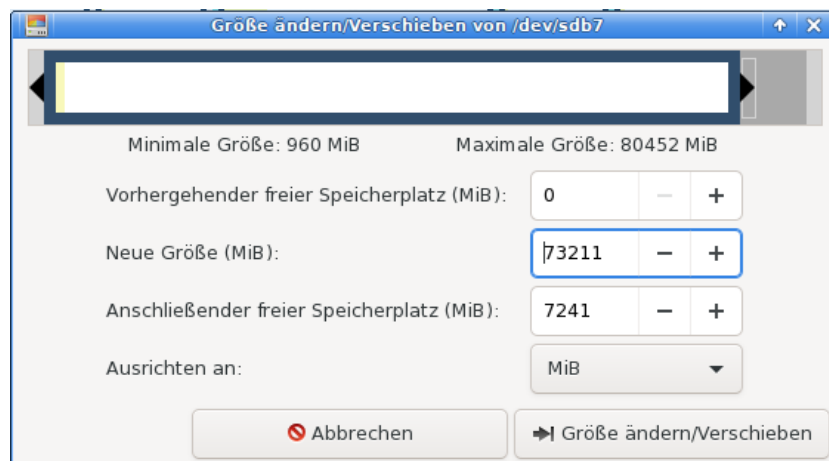
- **Eine neue Partition erstellen**

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.



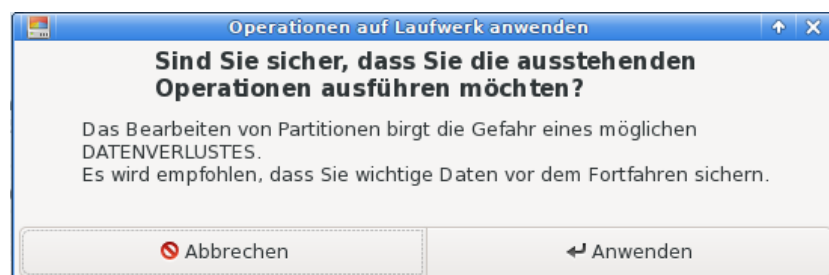
- **Falls ein Fehler gemacht wurde**

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



- **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.



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: 2026-02-22

## 4.7 Partitionieren mit fdisk

### ACHTUNG !

Das Anlegen von Partitionstabellen, Partitionen und das Bearbeiten von Partitionen vernichtet alle Daten auf dem betreffenden Datenträger.

Im Jahr 2000 begann die Einführung von GPT Partitionstabellen auf Basis des UEFI. Der neuere Standard **G**lobally **U**nique 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](#)

**fdisk** und das mit einer benutzerfreundlicheren ncurses Oberfläche versehene **cfdisk** erlauben die Bearbeitung von DOS Partitionstabellen auf Basis des BIOS und GPT Partitionstabellen auf Basis des UEFI.

Möchte man nicht nur eine vorhandene Partitionstabelle bearbeiten, sondern eine neue erstellen oder von DOS zu GPT wechseln, ist das Kommandozeilenprogramm **parted** die richtige Wahl. Ein kurzer Befehl erstellt auf **/dev/sda** eine neue GPT Partitionstabelle.

```
parted /dev/sda mktable gpt
```

Die Aktion ist noch einmal zu bestätigen, da alle bisher auf der Festplatte vorhandenen Daten verloren gehen.

### 4.7.1 Informationen zu Speichergeräten

Einige Informationen über die Geräte erhält man leicht von einem Pop-Up Fenster, wenn man auf dem Desktop mit der Maus auf das Icon eines Geräts geht. Dies funktioniert sowohl vom Live-ISO als auch bei einem installierten siduction. Ausführlichere Informationen bietet die Verwendung von **lsblk**, **blkid** und **fdisk** in einem root Terminal. Der Befehl

```
fdisk -l | tee /root/fdisk-l_$(date +%F_%H-%M-%S)
```

erzeugt neben dem unten gezeigten Auszug auch eine Datei mit Zeitstempel im Verzeichnis des Benutzers root. Dies kann sehr hilfreich sein, falls Probleme auftreten.

```
[...]
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: 0XXXXXXXXX

Device    Boot    Start        End    Sectors  Size Id Type
/dev/sda1                2048    83888127    83886080    40G 83 Linux
/dev/sda2    83888128    88291327    4403200    2,1G 82 Linux swap
/dev/sda4    88291328   312581807   224290480   107G  5 Extended
/dev/sda5    88293376   249774079   161480704    77G 8e Linux LVM
/dev/sda6   249776128   312581807    62803632    30G 8e Linux LVM

Disk /dev/nvme0n1: 465,76 GiB, 500107862016 bytes, 976773168 ↵
    sectors
Disk model: CT500P3SSD8
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: gpt
Disk identifier: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXX

Device      Start        End    Sectors  Size Type
/dev/sdb1      2048        206847    204800    100M EFI System
/dev/sdb2    206848   922953727   922746880    440G Linux filesystem
/dev/sdb3   922953728   976766975    53813248    25,7G Linux swap
```

## DOS Partitionstabelle

DOS Partitionstabellen gelten auf aktueller Hardware für Laptop und PC als veraltet und sollten dort nicht mehr verwendet werden. Auf kleineren Speichergeräten wie USB Stick oder Speicherkarte sind sie häufiger anzutreffen.

Die Partitionen in einer DOS Partitionstabelle können vom Typ *primär*, *erweitert* und *logisch* sein. Sie werden durch eine Zahl zwischen 1 und 15 definiert.

Es sind maximal vier primäre Partitionen anlegbar. Eine dieser Partitionen kann eine erweiterte sein. Innerhalb der erweiterten Partition wiederum sind bis zu elf logische Partitionen möglich. Somit ist die Anzahl der Partitionen auf 14 begrenzt.

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. Ihre Bezeichnungen beginnen mit Nummer 5 und enden mit Nummer 15.

### Beispiel

```
4 Partitionen, alles primäre:

|sda1|sda2|sda3|sda4|

6 mountbare Partitionen
  2 primäre, 1 erweiterte, darin 4 logische:

|sda1|sda2|-
           |
           |sda4| (erweiterte Partition)
           |
           |sda5|sda6|sda7|sda8|
```

/dev/sda5 kann nur eine logische Partition sein (in diesem Fall die erste logische auf diesem Gerät).

### GPT Partitionstabelle

Im Gegensatz zu der DOS Partitionstabelle erlaubt die GPT Partitionstabelle Platten/Partitionen größer als 2 TByte und eine theoretisch unbegrenzte Anzahl primärer Partitionen. Es gibt keine erweiterte Partition und auch keine logischen Partitionen.

Das Beispiel in dem folgenden Kapitel *Cfdisk verwenden* beruht auf einer Festplatte mit GPT Partitionstabelle.

### 4.7.2 Cfdisk verwenden

#### Daten zuvor sichern!

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

Das Programm cfdisk bitte nur auf einer Festplatte anwenden, deren sämtliche Partitionen nicht eingehangen sind.

Wir starten cfdisk in einer root Konsole (nach **su** ist die Eingabe des root Passworts gefordert).

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

#### Die Bedienoberfläche

Im ersten Bildschirm zeigt cfdisk die aktuelle Partitionstabelle. Am unteren Ende des Fensters befinden sich die 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.

```

Disk: /dev/sdc
Size: 232,89 GiB, 250059350016 bytes, 488397168 sectors
Label: gpt, identifier: 564

Device      Start      End      Sectors   Size Type
>> Free space 2048      488397134 488395087 232,9G

Filesystem UUID: 0BCB-1D55
Filesystem: vfat

[ New ] [ Quit ] [ Help ] [ Write ] [ Dump ]

Create new partition from free space
```

Die Festplatte */dev/sdc* enthält nur eine leere GPT Partitionstabelle.

#### Erstellen einer neuen Partition

Im ersten Schritt legen wir eine 300 MB große Efi System Partition (ESP) an. Der bereits markierte Befehl **New** wird mit der **Enter** Taste bestätigt.

```

Disk: /dev/sdc
Size: 232,89 GiB, 250059350016 bytes, 488397168 sectors
Label: gpt, identifier: 56

Device      Start      End      Sectors   Size Type
>> /dev/sdc1 2048      616447   614400    300M Linux filesystem
Free space  616448    488397134 487780687 232,6G

Partition UUID: D940C2BA-B9B2-4516-88C1-E0B6EBE848EA
Partition type: Linux filesystem (0FC63DAF-8483-4772-8E79-3D69D84
Filesystem UUID: 0BCB-1D55
Filesystem: vfat

[ Delete ] [ Resize ] [ Quit ] [ Type ] [ Help ]
[ Write ] [ Dump ]

Change the partition type
```

Damit das UEFI die ESP erkennt muss ihr Partitonstyp auf *EFI System* gesetzt werden. Mit dem Befehl **Type** fahren wir fort.

```

Select partition type
EFI System
MBR partition scheme
Intel Fast Flash
BIOS boot
Sony boot partition
Lenovo boot partition
PowerPC PreP boot
ONIE boot
ONIE config
Microsoft reserved
Microsoft basic data
Microsoft LDM metadata
Microsoft LDM data
Windows recovery environment
IBM General Parallel Fs
Microsoft Storage Spaces

C12A7328-F81F-11D2-BA4B-00A0C93EC93B
```

Die Pfeiltasten führen uns zu *EFI System*. Am unteren Rand wird die GUID des Partitonstyps angezeigt. Die **Enter** Taste schließt die Auswahl ab.

In den nächsten Schritten legen wir mit der gleichen Vorgehensweise drei weitere Partitionen an.

1,6G Typ: Linux extended boot

18G Typ: Linux swap

213G Typ: Linux filesystem

Das Ergebnis:

```
Disk: /dev/sdc
Size: 232,89 GiB, 250059350016 bytes, 488397168 sectors
Label: gpt, identifier: 56

Device      Start      End      Sectors   Size Type
/dev/sdc1    2048       616447   614400    300M EFI System
/dev/sdc2    616448     3893247  3276800    1,6G Linux extended boot
/dev/sdc3    3893248    41641983 37748736    18G Linux swap
>> /dev/sdc4 41641984 488396799 446754816 213G Linux filesystem

Partition UUID: BF625936-DB74-442A-88BA-2079B39C4079
Partition type: Linux filesystem (0FC63DAF-8483-4772-8E79-3D69D847)

[ Delete ] [ Resize ] [ Quit ] [ Type ] [ Help ]
[ Write ] [ Dump ]

Quit program without writing changes
```

Für unser siduction haben wir jetzt eine 213 GB große Partition. Das ist sinnvoll, wenn wir LVM oder das Dateisystem Btrfs nutzen möchten. Eine andere Variante wäre eine Systempartition mit 60 GB und eine weitere Partition mit 153 GB für davon unabhängige Daten. Beide erhalten später das Dateisystem ext4.

### Größe einer Partition ändern

Die Partition `/dev/sdc4` wird markiert und der Befehl **Resize** ausgewählt und bestätigt.

```
Disk: /dev/sdc
Size: 232,89 GiB, 250059350016 bytes, 488397168 sectors
Label: gpt, identifier: 564

Device      Start      End      Sectors   Size Type
/dev/sdc1    2048       616447   614400    300M EFI System
/dev/sdc2    616448     3893247  3276800   1,6G Linux extended boot
/dev/sdc3    3893248    41641983 37748736   18G Linux swap
>> /dev/sdc4 41641984 488396799 446754816 213G Linux filesystem

Partition UUID: BF625936-0B74-442A-88BA-2079B39C4079
Partition type: Linux filesystem (0FC63DAF-8483-4772-8E79-3D69D847)
New size: 60G
May be followed by M for MiB, G for GiB, T for TiB, or S for sectors.
```

Wir ändern die Größe auf 60G. Wieder beendet die **Enter** Taste den Vorgang.

Anschließend erstellen wir mit dem gesamten Rest (153 GB) eine Partition vom Typ *Linux filesystem*.

Die Partitionierung ist jetzt abgeschlossen.

### Partitionstabelle schreiben

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

```
Disk: /dev/sdc
Size: 232,89 GiB, 250059350016 bytes, 488397168 sectors
Label: gpt, identifier: 564

Device      Start      End      Sectors   Size Type
/dev/sdc1    2048       616447   614400    300M EFI System
/dev/sdc2    616448     3893247  3276800   1,6G Linux extended boot
/dev/sdc3    3893248    41641983 37748736   18G Linux swap
/dev/sdc4    41641984 167471103 125829120  60G Linux filesystem
>> /dev/sdc5 167471104 488396799 320925696 153G Linux filesystem

Partition UUID: 3302B555-D2E7-4F24-B186-F2C1BECB9775
Partition type: Linux filesystem (0FC63DAF-8483-4772-8)
Are you sure you want to write the partition table to disk? yes
Type "yes" or "no", or press ESC to leave this dialog.
```

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

## 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.

### 4.7.3 Formatieren von Partitionen

Es gibt für Linux verschiedene Dateisysteme, die man benutzen kann. Da wären **Ext2, Ext4, Btrfs, 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/sdXX**. Für "XX" trägt man die Bezeichnung der ausgewählten Partition ein.

```
mkfs.ext4 /dev/sdc4  
[...]  
mkfs.ext4 /dev/sdc5
```

Nach Abschluss der Formatierung erscheint eine Meldung über den Erfolg der Aktion. Ist das nicht der Fall, so ist bei der Partitionierung etwas schief gelaufen oder die Partition ist keine Linux Partition. Wir überprüfen das mit:

```
fdisk -l /dev/sdc
```

Gegebenenfalls muss noch einmal partitioniert werden.

War die Formatierung erfolgreich, wiederholen wir diesen Ablauf für die anderen Partitionen, wobei der Befehl an die Partition und das gewünschte Dateisystem anzupassen ist. (z. B.: **mkfs.ext2** oder **mkfs.fat** oder **mkfs.btrfs** usw.)

Bitte die Manpage **man mkfs** lesen.

Unserem Beispiel folgend erhalten die Partitionen /dev/sdc1 und /dev/sdc2 ein fat Dateisystem vom Typ 32.

```
mkfs.fat -F 32 /dev/sdc1  
[...]  
mkfs.fat -F 32 /dev/sdc2
```

Zuletzt wird die Swap Partition formatiert, in diesem Fall /dev/sdc3:

```
mkswap /dev/sdc3
```

Im Anschluss wird die Swap Partition aktiviert:

```
swapon /dev/sdc3
```

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/sdc3	partition	37748736	0	-2

Anschließend informieren wir den Kernel über die Änderungen mit dem Befehl **systemctl daemon-reload**.

Jetzt kann die Installation beginnen.

Zuletzt bearbeitet: 2026-02-21

## 4.8 LVM-Partitionierung - Logical Volume Manager

**Es folgt nun eine Basiseinführung.** Es liegt am geschätzten Leser, sich tiefer in die Materie einzuarbeiten. Bitte beachten sie die jeweilige man-Page. 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) [PV]:** 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) [VG]:** 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) [LV]:** Logische Volumen werden innerhalb 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.

#### 4.8.1 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 wird `cfdisk` benötigt, da zur Zeit GParted bzw. der KDE-Partitionsmanager (`partitionmanager`) das Anlegen von *Logical Volumes* nicht unterstützen. Siehe auch die Handbuchseite:

[Partitionieren mit `cfdisk` \(msdos-MBR\)](#)

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
    "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 [PV]

```
pvcreate /dev/sda1
```

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

## 3. Erstellen einer Volumengruppe [VG]

Nun erzeugen wir die VG mit dem Namen **vulcan** und ordnen ihr drei PV zu (in unserem Beispiel die Laufwerke /dev/sda1 /dev/sdb1 /dev/sdc1):

```
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 [LV]

An dieser Stelle muss entschieden werden, wie groß das LV 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 LV mit dem Namen **spock** innerhalb der VG Namens **vulcan**:

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

## 5. Formatieren des LV

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

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

## 6. Einbindung des LV

Erstellen des Mountpoints mit

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

Um das LV 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 LV kann geändert werden, sodass andere Nutzer Lese- bzw. Schreibzugang erhalten:

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

Die Schritte 4 bis 6 wiederholen wir nun für das weitere LV **kirk**.

Ein einfacher LVM sollte nun erstellt sein.

### 4.8.2 Größenänderung eines Volumens

Achtung

Immer zuerst eine Datensicherung vornehmen.

Wir empfehlen die Verwendung einer Live-ISO, für die Größenänderung. Obwohl die Vergrößerung eines Volumens im laufenden System ohne Fehler durchgeführt werden kann, ist dies bei der Verkleinerung nicht der Fall. Anomalien können zu einem Datenverlust führen, vor allem wenn die Verzeichnisse `/` (root) oder `/home` betroffen sind.

### Beispiel einer Vergrößerung

Das LV `spock` soll von 300GB auf 500GB vergrößert werden.  
Wir prüfen zuerst, ob genug freier Speicherplatz vorhanden ist.  
`vgdisplay` gibt Auskunft.

```
vgdisplay vulcan
[...]  
Free PE / Size      170890 / 667,50 GiB  
[...]
```

Es ist genug freier Speicherplatz für unser Vorhaben vorhanden.  
Wir können mit der Arbeit beginnen.  
Aushängen des LV:

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

#### Achtung

Hängen Sie nie Ihr root-Dateisystem im laufenden Betrieb aus.

Erweitern des LV und seines Dateisystems:

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

Dem Befehl `lvextend` ist als Option der Wert für die **Größenänderung** anzugeben und nicht die gewünschte Gesamtgröße.

Anschließend wird mit der Option `--resizefs` zuerst das Dateisystem überprüft und dann an die neue Größe des LV angepasst.

Zuletzt hängen wir das LV wieder ein.

```
mount /media/spock
```

Die Vergrößerung eines LV, sogar für das “/” (root) Dateisystem, ist auch im laufenden Betrieb möglich. Lediglich die beiden Befehle `umount` und `mount` entfallen. Allerdings wird dann kein Dateisystemcheck durchgeführt.

Wenn Sie das root-Dateisystem überprüfen wollen, nutzen Sie den Kernel-Kommandozeilen-Parameter `fsck.mode=force` beim Boot-Vorgang.

### Beispiel einer Verkleinerung

Das LV `spock` 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 LV 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 “`resize2fs`”-Befehl passt das Dateisystem exakt an die Größe des LV an.

### 4.8.3 LVM mit einem GUI-Programm verwalten

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

### 4.8.4 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)

Zuletzt bearbeitet: 2026-02-22

## 4.9 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
...
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
```

#### 4.9.1 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 2020 ..
-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-error
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.

### 4.9.2 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:

```
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](#) ersetzt den WPA-Supplicant. Seine 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 Kommandozeile 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
```

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
```

### 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 --
```

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

```
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
[...]
```

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

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



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 <Schnittstellename> disconnect`.

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.

In seltenen Fällen treten mit iwd Verbindungsabbrüche auf. Dann bietet es sich an [zum WPA-SupPLICant zurück](#) zu kehren.

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 Graphische 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 selbst-erklärend sein!
- **conman** ist ein von Intel entwickelter Netzwerkmanager, klein und Ressourcen schonend. 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   
Wlrkl1chS3hrG3h31m 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 Eingabeprompt `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 Quellen 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
# dhcpcd 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/NetworkManager/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: 2023-11-09

## 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 eingegeben:  
`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 (alles in einer Zeile):

```
//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 (alles in einer Zeile):

```
//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

### SSH aktivieren

Bei siduction ist ssh sowohl auf dem Live-ISO als auch nach der Installation nicht aktiviert!

Um ssh zu aktivieren und deaktivieren nutzen Sie bitte die Scripte `sshactivate` und `sshdeactivate`. Sie befinden sich in `/usr/sbin`.

Alternativ nutzen sie die Starter im Anwendungsmenü > Internet/Netzwerk.

### 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.

Generell sollte SSH zur Remote Steuerung nur konfiguriert und eingesetzt werden, wenn man sich sicher ist, welche Auswirkungen die Einstellungen in der Datei `/etc/ssh/sshd_config` und ggf. den Dateien im Verzeichnis `/etc/ssh/sshd_config.d/` haben.

Im Zweifel ist die Dokumentation in den Manpages `man sshd_config`, `man ssh_config` oder in Internet zu studieren, oder der Rat fachkundiger Personen einzuholen.

### 5.5.1 SSH absichern

Es ist nicht sicher, Root-Anmeldungen via SSH zum Standard zu machen. Debian sollte sicher sein, nicht unsicher. Ebenso sollen Angreifer nicht die Möglichkeit haben, über einen längeren Zeitraum einen wortlistenbasierten Passwort

Angriff (brute force attack) auf den SSH-Login durchzuführen. Deshalb nehmen wir einige Änderungen in der Datei `/etc/ssh/sshd_config` vor.

**Folgende Einstellungen können zur Erhöhung der Sicherheit angepasst werden:**

(Um die Funktion der Einträge zu aktivieren muss das Kommentarzeichen '#' entfernt werden.)

- **Port :**

Dieser Eintrag muss auf den Port verweisen, der auf dem Router zur Weiterleitung freigeschaltet ist. Die IANA hat dem Protokoll den TCP-Port 22 zugeordnet und Debian setzt ihn als Standard. Es ist jedoch ratsam einen Port außerhalb des Standardscanbereichs zu verwenden. Wir benutzen deswegen z.B. Port 5874 um Angriffe zu erschweren, da der SSH-Port dem Angreifer nicht bekannt ist.

```
Port 5874
```

- **ListenAddress :**

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
```

Das SSH Protokoll 2 mit seinen verbesserten und erweiterten Funktionen ist bereits Grundeinstellung bei Debian.

- **LoginGraceTime :**

Die erlaubte Zeitspanne beträgt als Standard 2 Minuten. Da man für gewöhnlich keine zwei Minuten benötigt, um Benutzernamen und Passwort einzugeben, stellen wir eine etwas kürzere Zeitspanne ein:

```
LoginGraceTime 30
```

Nun hat man 30 Sekunden Zeit zum Anmelden, und Hacker haben keine zwei Minuten bei jedem Versuch, das Passwort zu knacken.

- **PermitRootLogin <prohibit-password>:**

Warum Debian hier Erlaubnis zur Anmeldung als Root erteilt, ist nicht nachvollziehbar. Wir korrigieren zu `no`:

```
PermitRootLogin no
StrictModes yes
```

Alternativ kann man diese Option auf *forced-commands-only* setzen. Damit wird die Anmeldung von root über asymmetrische Authentifizierung erlaubt, aber nur falls die Option *command* festgelegt wurde (was nützlich für die Durchführung ferner Sicherungskopien ist, selbst wenn die Anmeldung von root normalerweise nicht erlaubt ist). Alle anderen Authentifizierungsmethoden für root bleiben deaktiviert.

- **MaxAuthTries :**

Mehr als 3 oder 4 Versuche sollten nicht ermöglicht werden:

```
MaxAuthTries 3
```

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

- **AllowUsers :**

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 :**

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 :**

natürlich muss hier 'yes' gesetzt werden. Es sei denn, man verwendet einen KeyLogin.

```
PasswordAuthentication yes
```

Schlussendlich:

```
systemctl restart ssh
```

Nun hat man eine etwas sicherere 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 libreoffice 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 erhält man mit **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 oder Thunar

Dolphin und Thunar sind fähig, auf Daten eines entfernten Rechners zuzugreifen, indem sie das *sftp* Protokoll benutzen, welches in ssh vorhanden ist.

So wird es gemacht:

1. **Für einen entfernten Server**

Man öffnet ein neues Dateimanager-Fenster oder einen neuen Reiter.

Eingabe in die Adress-Leiste nach dem Muster:

- `sftp://username@ssh-server.com`

Dann öffnet sich ein Dialog-Fenster und fragt nach dem SSH-Passwort. Man gibt das Passwort ein und klickt auf OK,

- oder die Eingabe enthält bereits das Passwort:

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

Es wird nicht nach einem Passwort gefragt, man wird direkt verbunden.

2. **Für eine LAN-Umgebung**

Nach dem gleichen Muster wie zuvor:

- `sftp://username@192.168.x.x`

mit Dialog-Fenster für das SSH-Passwort,

- oder gleich mit Passwort:

```
sftp://username:passwort@192.168.x.x
```

Bitte richtige IP eingeben!

Eine SSH-Verbindung ist nun hergestellt. In diesem 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. Die Eingabe folgt dann der Syntax:

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

dies ist auch 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 der Datei `/etc/fstab` (alles in einer Zeile):

```
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 Nutzernamen (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 Nutzernamen gelistet und folgender Befehl ausführbar sein:

```
mount lokaler_mountpunkt  
    und  
umount lokaler_mountpunkt
```

Zuletzt bearbeitet: 2023-10-19

## 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:

- **Linux**: Betriebssystem
- **Apache**: Web-Server
- **MariaDb**: Datenbank-Server (ab Debian 9 'Stretch', zuvor MySQL)
- **PHP**, 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” 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 autoremove
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 Datenbank-Benutzers für das Entwicklungsprojekt.
  - Das *Passwort* für den neuen Datenbank-Benutzer.
  - Den *Namen* (Login-Name) eines neuen Datenbank Administrators.
  - Das *Passwort* für den Datenbank Administrator.

### 5.6.1 Apache installieren

Die Installation des Webserver 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 Metapackete “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 “*mysql*” 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
[...]
```

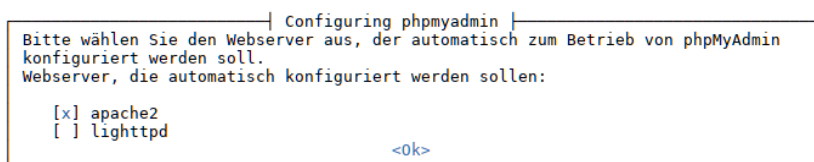
Die folgenden NEUEN Pakete werden installiert:

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-var-exporter  
php-symfony-service-contracts 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*



im zweiten, am Ende der Installation, wählen wir *“ja”* aus.

```
| Configuring phpmyadmin |
Für das Paket phpmyadmin muss eine Datenbank installiert und konfiguriert sein,
bevor es benutzt werden kann. Dies kann optional mit Hilfe von dbconfig-common
geschehen.
Falls Sie ein erfahrener Datenbankadministrator sind und wissen, dass Sie diese
Konfiguration manuell durchführen möchten oder, falls Ihre Datenbank bereits
installiert und konfiguriert ist, verwerfen Sie diese Option. Details zur
manuellen Installation sind üblicherweise in /usr/share/doc/phpmyadmin zu finden.
Andernfalls sollte diese Option wahrscheinlich gewählt werden.

Konfigurieren der Datenbank für phpmyadmin mit dbconfig-common?

<Ja>                                <Nein>
```

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 Webserver 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

```
# journalctl -n 20 -u mariadb.service
```

und

```
# journalctl -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.2
      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 irgendein anderes 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: 2025-09-19

## 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.

Verbindungsname: LAN

Allgemein Ethernet 802.1X-Sicherheit DCB Proxy IPv4-Einstellungen IPv6-Einstellungen

Methode: Manuell

Adressen

Adresse	Netzmaske	Gateway
192.168.3.1	24	

DNS-Server

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

```
$ 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 nutzen, da mit den Rechten dieser Gruppe Apache läuft.

Wir nennen die neue Gruppe **work**.

#### 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 work
adduser <user> work
chgrp work /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 work
```

2. SGID-Bit für "*DocumentRoot*" setzen, damit alle hinzukommenden Verzeichnisse und Dateien die Gruppe `work` 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 work 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 work 10701 9.Jan 19:04 index.html
-rw-r--r-- 1 root work  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 work  4096 9.Jan 19:32 .
drwxr-xr-x 3 root      root  4096 9.Jan 19:04 ..
-rw-rw---- 1 www-data work 10701 9.Jan 19:04 index.html
-rw-rw---- 1 www-data work   20 9.Jan 19:32 info.php
```

Jetzt haben in “*DocumentRoot*” nur Mitglieder der Gruppe `work` 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](http://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 enthält eine empfehlenswerte Seite mit diversen Tipps zur Absicherung.  
[apache.org - Security Tipps](https://httpd.apache.org/docs/2.4/security_tips.html) (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 aktiviert. 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]
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 Prompt **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.

### 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;`

### 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
#
```

## 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 system-relevante Daten erhält.

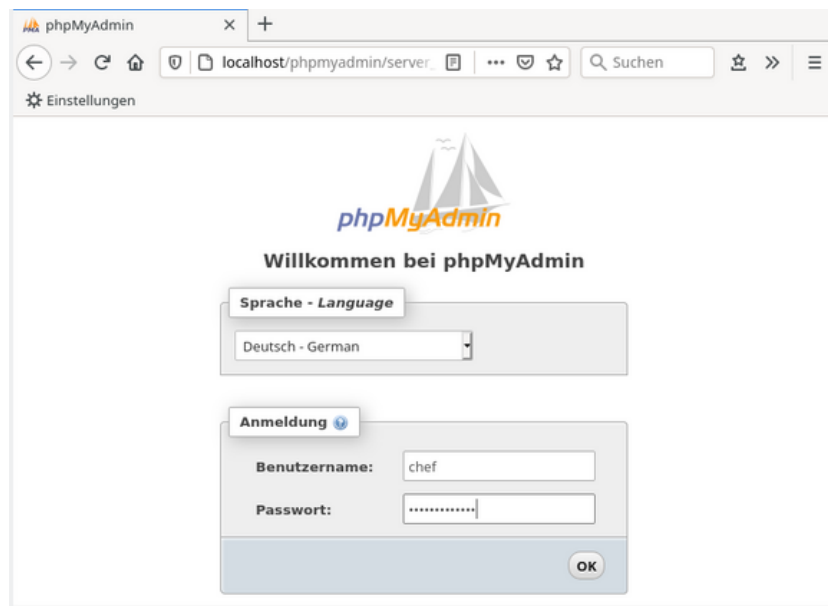
#### 5.8.4 phpMyAdmin

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.

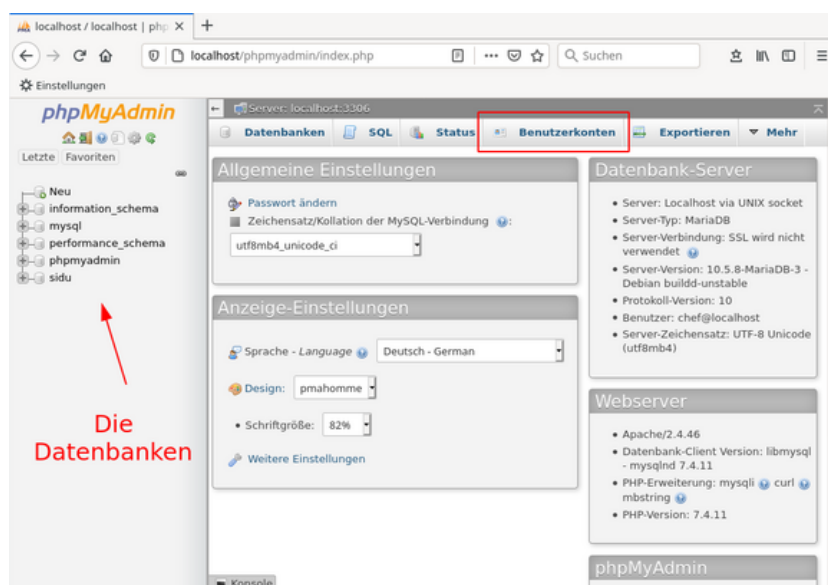
Wir verwenden das für weniger erfahrene Benutzer besser geeignete Programm **phpMyAdmin** und geben in die Adresszeile des Browsers `http://localhost/phpmyadmin/`

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.



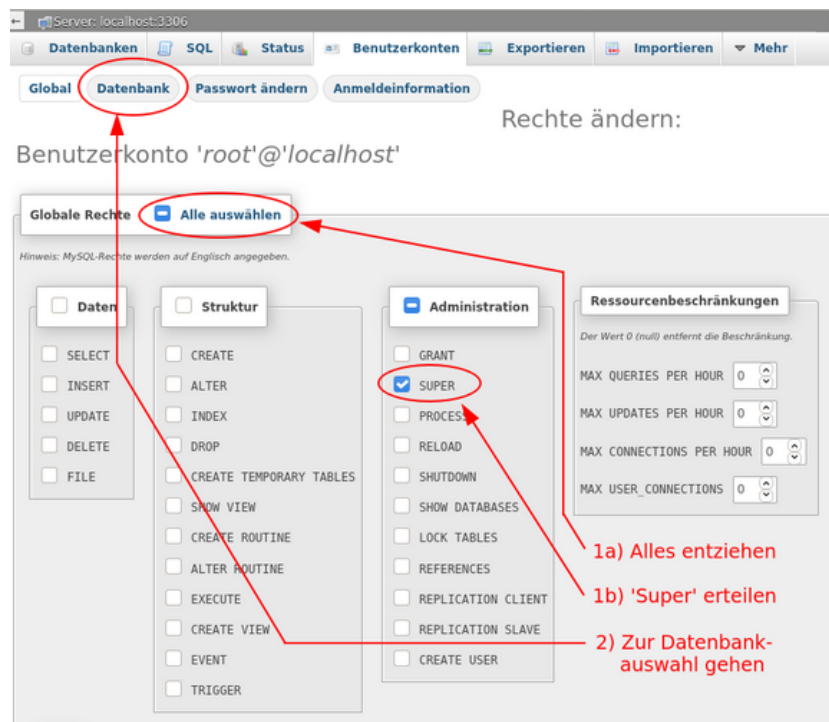
Im Startfenster sehen wir in der linken Spalte alle Datenbanken. Im Hauptteil wählen wir den Reiter **Benutzerkonten**.



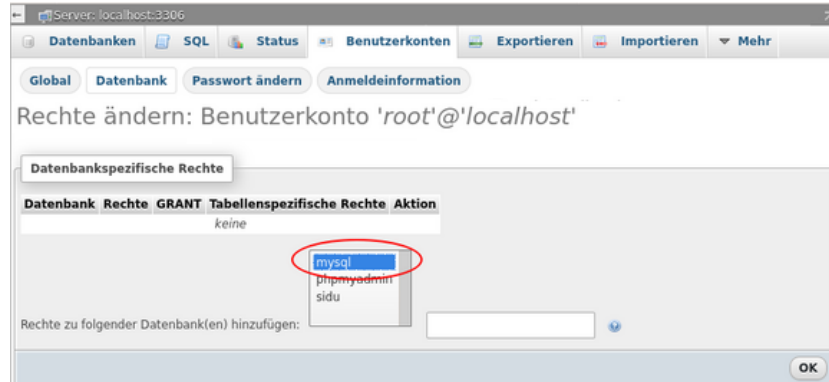
Die Benutzerkontenübersicht stellt alle Benutzer und in Kurzform deren Rechte dar. Wir wählen hier für den Benutzer **root** den Schalter **Rechte ändern**.



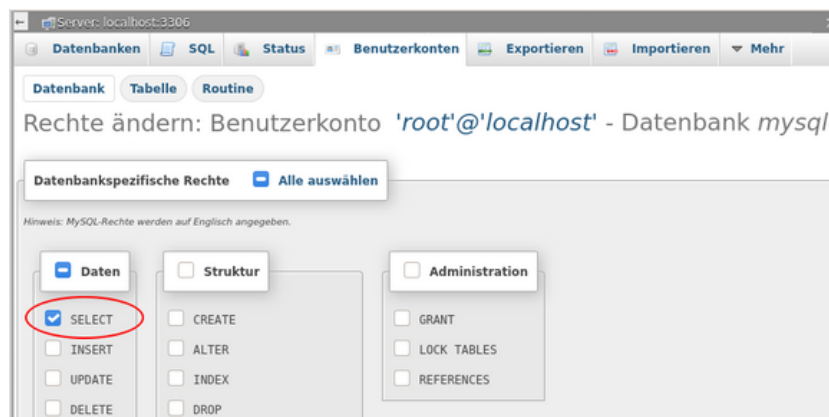
Nun sehen wir für den Benutzer **root** die detaillierten Rechte. Hier entziehen wir ihm erst einmal alle Rechte (1a), erteilen dann im Bereich **Administration** das Recht "Super" (1b) und führen die Aktion aus, indem wir ganz unten rechts auf dieser Seite den **OK**-Button anklicken (im Screenshot nicht sichtbar).



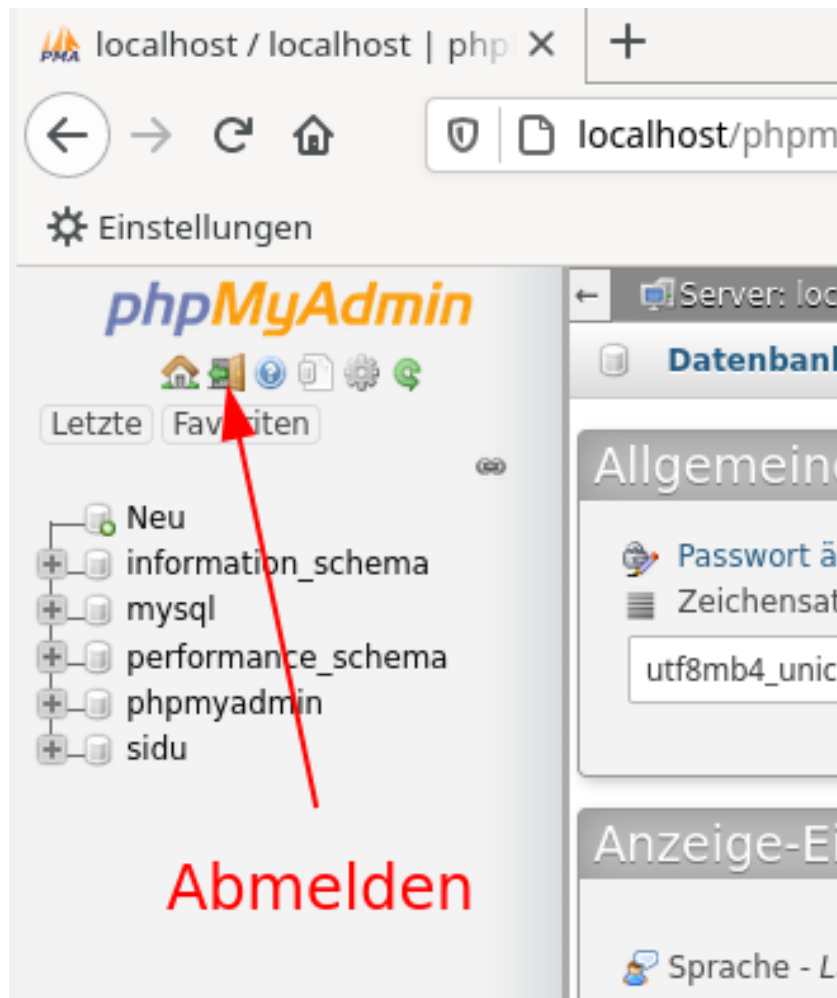
Anschließend gehen wir über den Datenbank-Schalter (2) zur nächsten Seite.



Nach Auswahl der Datenbank "mysql" und OK öffnet sich diesmal ein Fenster mit den detaillierten Rechten an der Datenbank "mysql" für den Benutzer root.



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.

### 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 Performance-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 Verzeichnisse 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-Skripte 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 Skripten 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 Interfaces 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 nVidia, 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.

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.d/debian.sources` ein Eintrag analog

```
Types:      deb
URIs:       https://deb.debian.org/debian/
Suites:      unstable
Components: main contrib non-free non-free-firmware
Enabled:     yes
Signed-By:   /usr/share/keyrings/debian-archive-keyring.gpg
```

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 Kernen, 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

(alles in einer Zeile)
```

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 empfiehlt 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: 2025-02-12

## 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.
- [Btrfs Dateisystem in siduction](#). Die Subvolumen nach der Installation. Neue Subvolumen anlegen und verwalten. Snapshot in Btrfs.
- [Btrfs Snapshots mit Snapper](#) erstellen und verwalten. Die Konfiguration von Snapper und die Zusammenarbeit mit systemd. System Rollback und Wiederherstellung von Dateien.
- [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.
- [Nala für das Paketmanagement](#), Ein Front-End, dass APT für den Anwender optimiert und beschleunigt.
- [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 Webserverns”.
- [systemd-timer Unit](#), die benötigten Dateien, die Optionen der timer-Unit, timer-Unit anlegen und eingliedern, timer-Unit als cron Ersatz.
- [systemd-boot](#), der Bootmanager von systemd. Besonderheiten, Vergleich mit Grub, Installation.
- [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: 2024-09-21

## 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”* eingibt.

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 Prompt) *\$* 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 Prompt 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 jedoch 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 Prompts 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.

```

Terminal - user1@pc1:/home/user1/Dokumente
Datei Bearbeiten Ansicht Terminal Reiter Hilfe
user1@pc1:~/Dokumente$ su
Passwort:
root@pc1:/home/user1/Dokumente#

```

Der Fokus beim Arbeiten mit dem Terminal sollte auf den Eingaben und Ausgaben der Befehle liegen und nicht auf bunte Prompts. 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 Prompts ändern

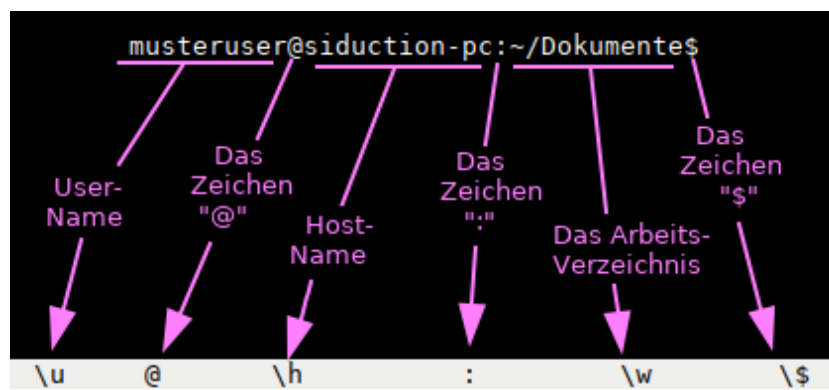
Bevor die Konfigurationsdatei geändert wird, erstellen wir im Terminal erst eine Sicherungskopie mit einem Datumsstempel.

```
$ 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\z
\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '
```

Dem Schrift- und Farbcode folgen unmittelbar die Promptteile, die diese Darstellung erhalten sollen. In der folgenden Abbildung wird der Bezug zwischen den Promptteilen und ihren Kürzeln gezeigt.



Die anschließende Tabelle erklärt die Werte der Syntax `"\033[01;32m"`, 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	Promptteile und ihre Darstellung
[01;32m]\u@\h	user, @ und host erhalten die Attribute "fett" und "grün"
[00m]:	Doppelpunkt erhält die Standardattribute des Terminals
[01;34m]\w	das Arbeitsverzeichnis erhält die Attribute "fett" und "blau"
[00m]\\$	der Prompt erhält die Standardattribute des Terminals

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 Prompts die Farbkodierung anzupassen.

```
PS1='${debian_chroot:+($debian_chroot)}\[\033[03;32m\]\u@\h\z  
  \[\033[01;34m\]:\w\[\033[00m\]\$ '
```

Dieses Codebeispiel erzeugt einen Prompt, 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 Terminal in einem abnormalen Zustand zurückgelassen hat. Dann muss

```
reset
```

einggegeben 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 aufgebaut. Solche Verzerrungen treten meist auf, wenn man mit Programmen arbeitet, die eine ncurses-Schnittstelle benutzen, zum Beispiel *“cfdisk”*.

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](http://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 Dateinamens 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 Prompt 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: 2026-02-22

## 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 Hierarchie, 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.

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. <a href="#">Auf dieser Unit wird eine Systemaktualisierung (dist-upgrade) durchgeführt</a> .
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:

```
$ pgrep <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 harmonisieren 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.

$1680 \text{ Px} \times 25,4 \text{ mm/inch} : 474 \text{ mm} = 90 \text{ Px/inch (dpi)}$

- 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 / Antialiasing” :*

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 Antialiasing für LCD-Bildschirme, indem zusätzlich die Farbkomponenten eines Pixels angesteuert werden.

*“Hinting” :*

Ist die Anpassung (Veränderung) der Schriftzeichen an das Pixelraster des Bildschirms. Dadurch verringert sich der Bedarf an Antialiasing, 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	184
27 Zoll	598 mm	336 mm	163
28 Zoll	620 mm	349 mm	157
32 Zoll	708 mm	398 mm	138
37 Zoll	819 mm	461 mm	119
42 Zoll	930 mm	523 mm	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-usplix`. 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 Drop-down-Menü, das 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: 2025-09-19

## 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 as root cmd systemctl args isolate
deny lisa cmd systemctl args "isolate emergency.target"
deny lisa cmd systemctl args "isolate rescue target"
deny lisa cmd systemctl args "isolate graphical.target"
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* wechseln und nach Abschluss einen *systemctl reboot* durchführen. Die Regelzeile *“permit persist lisa as root cmd systemctl args isolate”* bewirkt, dass alle Aufrufe von *systemctl isolate* erlaubt sind, außer denen, die in den nachfolgenden Regeln verboten werden. Deshalb kann sie nicht direkt vom *multi-user.target* in das *graphical.target* wechseln. Hier sehen wir den Aufbau einer Hierarchie.

### Hinweise

Wer immer wieder *sudo* eintippt, dem hilft die Zeile *alias sudo="doas"* in sei-

ner `.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: 2025-09-19

## 7.4 Btrfs

Btrfs ist ein modernes Copy-on-Write Dateisystem für Linux.

siduction unterstützt die Installation in eine mit *Btrfs* formatierte Partition. Mit der Veröffentlichung von 2022.12.0 kommt die Möglichkeit hinzu, mit Snapper Snapshots von Btrfs zu verwalten und über Grub zu booten. Das Installationsprogramm legt dabei innerhalb der ausgewählten Partition Subvolumen für das Wurzelverzeichnis `@`, die Benutzerverzeichnisse `@home` und `@root`, das Verzeichnis `@var@log` sowie ein Subvolumen `@snapshots` für System Snapshots an.

Btrfs funktioniert gut mit SSDs und herkömmlichen Festplatten. Der eigene eingebaute RAID Mechanismus (unterstützt wird RAID 0, 1 und 10) arbeitet auch bei Festplatten verschiedener Größe zuverlässig. Metadaten und Dateidaten behandelt Btrfs unterschiedlich. Normalerweise werden Metadaten auch bei nur einem Laufwerk doppelt gespeichert. Bei mehreren Laufwerken kann der Administrator innerhalb des gleichen Dateisystems unterschiedliche RAID Level für die Metadaten und Dateidaten festlegen.

Btrfs verwaltet die Daten innerhalb der Laufwerke in Subvolumen, oberflächlich betrachtet ähnlich herkömmlichen Partitionen. Von den Subvolumen kann Btrfs Snapshots anfertigen, die bei Bedarf der Datenrekonstruktion dienen. Ein eingehängtes Btrfs Dateisystem verhält sich meistens wie jedes andere Linux-Dateisystem. Gelegentlich treten jedoch einige Unterschiede zutage, denn Btrfs erledigt seine Arbeit vorwiegend im Hintergrund. Für Verwirrung sorgt zum Beispiel das Löschen einer großen Datei, ohne dass sich sofort der verfügbare freie Speicherplatz erhöht. Einige Zeit später ist der fehlende Platz dann doch da, oder auch nicht, wenn ein vorangegangener Snapshot die Datei referenziert.

Zu Btrfs finden sich im Internet zahlreiche Dokumentationen. Wir werden deshalb hier nicht die umfangreichen Möglichkeiten sowie die Befehle und deren Anwendung wiederholen. Die Lektüre von `man btrfs` und `man btrfs-<Befehl>` ist obligatorisch. Darüber hinaus empfehlen wir das umfangreiche [Wiki von kernel.org](https://wiki.kernel.org/) und die ausführliche Dokumentation von [readthedocs.io](https://readthedocs.io/).

### Bitte beachten

siduction empfiehlt keine separate Boot-Partition bei Verwendung des Btrfs Dateisystems.

Das Verzeichnis `/boot` ist ein wesentlicher Bestandteil des Betriebssystems. Mit einer separaten Partition würde es von Systemsnapshots ausgenommen und damit ein Rollback möglicher Weise zu Fehlern führen.

Etwas differenzierter verhält es sich bei Verwendung von `systemd-boot`. In diesem Fall bitte die Handbuchseiten zu [systemd-boot](#) konsultieren.

### Btrfs verwenden

Für die fortschrittlichen Eigenschaften von Btrfs (Snapshots, Komprimierung, Defragmentierung, Selbstheilung für Daten und Metadaten, integrierte Datenträgerverwaltung ...) z.B. gegenüber ext4, benötigen wir erkennbar größere Laufwerke. Das ist derzeit meist kein Problem, denn selbst preiswerte PCs und Laptops verfügen oft über 500 GB große Laufwerke.

Als Mindestgröße des Btrfs Laufwerks, in das die vollständige Installation erfolgen soll, empfehlen wir 100 GB. Abhängig vom Volumen der privaten Daten auch deutlich mehr. Möchte man Btrfs nur für die root Partition verwenden, sollte diese eine Größe von mindestens 50 GB aufweisen. Für Benutzer die nicht so viel Speicherplatz zuweisen wollen, ist die übliche Vorgehensweise entweder Btrfs ohne Snapshots oder ext4 zu verwenden.

Btrfs versteht auf der Kommandozeile für seine Befehle und Optionen beliebige Abkürzungen, sofern diese eindeutig sind. So wird zum Beispiel `btrfs su li` / intern zu `btrfs subvolume list /`.

#### 7.4.1 Btrfs Subvolumen

Bei der Erstinstallation in eine einzige Partition werden die folgenden Subvolumen angelegt.

Subvolumen	Einhängpunkt	Bemerkungen
@	/	
@home	/home	
@root	/root	Der Benutzer <b>root</b>
@var@log	/var/log	
@snapshots	/.snapshots	Ablageort für die Snapshot von @
@home/.snapshots	-	Ablageort für die Snapshot von @home

Für Btrfs liegen sie, mit Ausnahme von `@home/.snapshots` gleichwertig auf der höchsten Ebene (*top level 5*). Es wird als *“flaches Layout”* bezeichnet. Lediglich im `@home` Subvolumen existiert eine Verschachtelung für die zugehörigen Snapshots. Die Dateisystemwurzel an sich wird nicht eingehängt, sondern die *top level 5* Subvolumen. Es ist nicht mehr nötig, das “Root”-Gerät einzuhängen, wenn nur der Inhalt der Subvolumen von Interesse ist. Im laufenden Betrieb befinden wir uns bereits in dem Subvolumen `@`.

Der Befehl `btrfs subvolume list /` gibt alle Subvolumen der Dateisystemwurzel aus. Die Option `-t` erstellt eine übersichtliche Liste.

```
# btrfs subvolume list -t /
ID    gen    top level  path
--    ---    -
256   22981   5         @
257   22952   5         @root
258   22982   5         @home
260   22967   5         @snapshots
269   22972   5         @var@log
271   22969   258       @home/.snapshots
```

### Das Standard Subvolumen

Während der Installation von siduction auf Btrfs wird das Subvolumen `@` als Btrfs Standard Subvolumen gesetzt, wie der folgende Befehl zeigt.

```
# btrfs subvolume get-default /
ID 256 gen 22981 top level 5 path @
```

Ein Rollback mit dem empfohlenen Befehl `snapper -a classic rollback <Nr>` setzt das Standard Subvolumen dann auf das Rollbackziel. Ausführlich behandelt wird ein Rollback etwas später im Kapitel [Snapper rollback](#).

### Neues Subvolumen anlegen

Wir wünschen uns ein zusätzliches Subvolumen mit dem Namen `@data`. Dadurch, dass ein Standard Subvolumen gesetzt wurde, bleibt die *top level 5* Ebene des Btrfs im Verzeichnisbaum verborgen. Nötig ist die zusätzlichen Mountoption `subvolid=5`. Mit ihr erreichen wir diese Ebene und können das neue Subvolumen darin anlegen. Abschließend hängen wir `/mnt` wieder aus.

```
# mount -t btrfs -o subvolid=5 /dev/sdxX /mnt/  
# btrfs subvolume create /mnt/@data  
Create subvolume '/mnt/@data'  
# ls /mnt/  
@ @data @home @root @snapshots @var@log  
# umount /mnt
```

Der `ls` Befehl zeigt nur die *top level* 5 Subvolumen, auch das neue.

Den Einhängepunkt des neuen Subvolumens legen wir im Wurzelverzeichnis an. Damit die normalen Benutzer darauf zugreifen können, ändern wir die Gruppe.

```
# cd /  
# mkdir /data  
# chgrp users /data
```

Diese wenigen Befehle reichen aus um ein zusätzliches *top level* 5 Subvolumen anzulegen und allen Mitgliedern der Gruppe *users* den Zugriff darauf zu ermöglichen.

Subvolumen lassen sich auch verschachteln und somit innerhalb bestehender Subvolumen erstellen. Wir raten zur besseren Übersicht eher zu einem flachen Schema. Eine Ausnahme bilden die von Snapper benötigten “.snapshots” Subvolumen.

### Subvolumen einhängen

Die Datei `/etc/fstab` enthält nach der Installation bereits alle notwendigen Einträge um die Subvolumen automatisch einzuhängen. Das Standard Subvolumen wird dabei immer, auch nach einem Rollback, unter `/` eingehangen.

Um zu zeigen, wie ein Subvolumen manuell eingehangen und die Datei `/etc/fstab` erweitert wird, benutzen wir das zuvor erstellte Subvolumen `@data`.

Mit dem Befehl

```
mount -t btrfs -o subvol=@data,defaults /dev/sdxX /data/
```

hängen wir das Subvolumen manuell ein.

Diese einfache Variante eignet sich nicht für eine dauerhafte Verwendung. Außerdem unterdrückt sie die vorteilhaften Fähigkeiten von Btrfs. Wir schauen uns einen Eintrag aus der Datei `/etc/fstab` an.

```
# grep home /etc/fstab
UUID=<hier> /home btrfs subvol=/@home,defaults,noatime,↵
    space_cache=v2,autodefrag,compress=zstd 0 0
```

Mit der Option “*space\_cache=v2*” werden die Adressen der freien Blöcke des Laufwerks zwischengespeichert, um die Schreibvorgänge zu beschleunigen. Die Option “*autodefrag*” sorgt für die Defragmentierung der Dateien während der Laufzeit und die letzte Option “*compress=zstd*” komprimiert die Daten.

Unser selbst erstelltes Subvolumen `@data` soll automatisch und dauerhaft mit diesen Optionen verfügbar sein. Deshalb ergänzen wir die Datei `/etc/fstab` um den benötigten Eintrag mittels zweier Befehle. Anschließend benutzen wir `systemd` um den Kernel über die Änderung zu informieren und hängen das neue Subvolumen zuletzt ein.

```
# echo -e "#\n# Extended by root on $(date +%F)" >> /etc/↵
    fstab
# grep home /etc/fstab | sed 's!home!data!g' "$@" >> /etc/↵
    fstab
# systemctl daemon-reload
# mount /data
```

Nun wird das Subvolumen `@data`, wie alle anderen auch, bei jedem Bootvorgang eingehangen.

### 7.4.2 Btrfs Snapshot

Ein Snapshot ist ein Subvolumen wie jedes andere, jedoch mit einem vorgegebenen Anfangsinhalt. Im Dateimanager betrachtet scheint es eine vollständige Kopie des ursprünglichen Subvolumens zu enthalten. Btrfs ist ein Copy-on-Write Dateisystem, sodass es nicht notwendig ist alle Daten tatsächlich zu kopieren. Der Snapshot hat einfach einen Verweis auf die aktuelle Wurzel des Dateisystems seines ursprünglichen Subvolumens. Erst wenn eine Datei im Subvolumen geändert wird, erstellt Btrfs eine Kopie der ursprünglichen Datei im Snapshot. In umgekehrter Richtung haben Änderungen einer Datei in einem Snapshot keine Auswirkung auf die Datei im ursprünglichen Subvolumen.

Ein Snapshot ist nicht rekursiv. Ein Subvolumen oder ein Snapshot ist effektiv eine Barriere. Dateien in verschachtelten Subvolumen erscheinen nicht im Snapshot. Stattdessen gibt es ein Blind-Subvolumen, was bei verschachtelten Layouts für Verwirrung sorgen könnte. Das nicht rekursive Verhalten erklärt, weshalb siduction während der Installation zusätzliche Subvolumen angelegt hat. So gelangen keine privaten und variablen Daten aus den Subvolumen `@home`, `@root` und `@var@log` in einen Snapshot von `@`.

Man sollte beachten, dass Snapshots von Btrfs Dateisystemen in keinem Fall eine durchdachte Datensicherung ersetzen. Selbst bei RAID1 und RAID10 Systemen mit Btrfs steht die Ausfallsicherheit im Vordergrund und nicht die Datensicherung.

### Snapshot erstellen

#### Achtung

Nur anwenden, wenn Sie Snapper **nicht** verwenden wollen.

Da ein Snapshot ein Subvolumen innerhalb seiner Quelle ist, bietet es sich an, ein entsprechendes Unterverzeichnis anzulegen. Wir nehmen für das Beispiel unser selbst erstelltes Subvolumen `@data`, legen das Verzeichnis an und gleich anschließend den ersten Snapshot.

```
# mkdir /data/.snapshots
# btrfs subvolume snapshot -r /data/ /data/.snapshots/01
Create a readonly snapshot of '/data' in '/data/.snapshots/01'
```

Der Befehl erinnert von der Syntax her an einen einfachen Kopiervorgang, wobei `01` der Ordner ist, in dem sich die Dateien des Snapshot befinden.

Statt `01` kann man `$(date +%F_%H-%M)` verwenden um das Datum und die Uhrzeit als Ordnernamen zu erhalten.

Standardmäßig werden Snapshots mit Lese- und Schreibzugriff erstellt. Mit der Option `-r` sind sie schreibgeschützt. Wir raten dringend die Option `-r` zu verwenden, denn ein Snapshot bildet zum Zeitpunkt seiner Erstellung den Zustand des Subvolumens ab. Wie man auf die Daten eines Snapshots zugreifen kann erfahren wir im Handbuch in den Kapiteln ab [“Snapper Rollback”](#).

## 7.5 Snapper

Snapper ist ein Werkzeug für die Verwaltung von Dateisystem Snapshots unter Linux für Btrfs und für thin-provisioned LVM Volumen. Neben der Erstellung und Löschung von Snapshots kann es auch Snapshots vergleichen und Unterschiede zwischen Snapshots rückgängig machen. Es ermöglicht Benutzern ältere Versionen von Dateien einzusehen und Änderungen rückgängig zu machen. Außerdem unterstützt Snapper automatische Snapshots nach Zeitplänen oder zu Aktionen.

Die Standardkonfiguration von Snapper in siduction umfasst automatische Pre- und Post-Snapshots des Subvolumen `@` bei Änderungen am System und die Vorbereitung von zeitgesteuerten Snapshots für beliebige andere Subvolumen. Außerdem wird eine zusätzliche Konfigurationsvorlage mit dem Namen *rolling* ausgeliefert.

Die Snapper Dateien befinden sich in:

- `/usr/bin/` Das ausführbare Programm `snapper`.
- `/usr/lib/snapper/` Hilfsprogramme für Snapper.
- `/etc/default/snapper` Eine Übersicht der konfigurierten Subvolumen.
- `/etc/snapper/configs/` Die Konfigurationsdateien der konfigurierten Subvolumen.
- `/usr/share/snapper/config-templates/` Die Konfigurationsvorlagen.
- `/var/log/snapper.log` Snappers Logdatei.

Bitte die man pages `man snapper` und `man snapper-configs` lesen.

### 7.5.1 Snapper Konfiguration

Snapper arbeitet mit systemd zusammen. Einige Einstellungen zur Handhabung der automatischen Snapshots verbergen sich in den zugehörigen systemd Units. Das Kapitel "[Snapper und systemd](#)" erklärt die Funktionen und gibt Hinweise zu deren Anpassung.

Siduction erstellt bei der Installation automatisch die Konfigurationen für die Subvolumen `@` und `@home`. Für andere Subvolumen müssen wir bei Bedarf selbst Konfigurationen nach dem folgenden Muster erstellen.

```
# snapper -c <config_name> create-config -t <config_vorlage> ↵
    <subvolume_mount_point>
```

Doch zuvor schauen wir uns die Konfiguration für das Subvolumen @ mit dem Namen `root`, @home mit dem Namen `home` und die beiden Vorlagen `default` von snapper sowie `rolling` von siduction an.

#### Snapper Konfiguration

Subvolumen	@	@home	--	--
conf-name or templ-name	root	home	default	rolling
Schlüssel	Wert	Wert	Wert	Wert
ALLOW_GROUPS	users	users		users
ALLOW_USERS				
BACKGROUND_COMPARISON	yes	yes	yes	yes
EMPTY_PRE_POST_CLEANUP	yes	yes	yes	yes
EMPTY_PRE_POST_MIN_AGE	1800	1800	1800	1800
FREE_LIMIT	0.2	0.2	0.2	0.2
FSTYPE	btrfs	btrfs	btrfs	btrfs
NUMBER_CLEANUP	yes	yes	yes	yes
NUMBER_LIMIT	50	50	50	20
NUMBER_LIMIT_IMPORTANT	10	10	10	5
NUMBER_MIN_AGE	1800	1800	1800	1800
QGROUP				
SPACE_LIMIT	0.5	0.5	0.5	0.5
SUBVOLUME	/	/home	/	
SYNC_ACL	yes	yes	no	no
TIMELINE_CLEANUP	yes	yes	yes	yes
TIMELINE_CREATE	no	no	yes	yes
TIMELINE_LIMIT_DAILY	10	10	10	6
TIMELINE_LIMIT_HOURLY	10	10	10	11
TIMELINE_LIMIT_MONTHLY	10	10	10	
TIMELINE_LIMIT_WEEKLY	0	0	0	1
TIMELINE_LIMIT_YEARLY	10	10	10	
TIMELINE_MIN_AGE	1800	1800	1800	1800

-----+-----+-----+-----+

Vom Subvolumen @ wird bei jeder APT Aktion ein “pre” und “post” Snapshot erstellt. Der Schlüssel `NUMBER_LIMIT=50` bewirkt, dass die jüngsten fünfundzwanzig Snapshotpaare erhalten bleiben.

siduction hat den `ALLOW_GROUPS=users` gesetzt und ermöglicht so allen Mitgliedern der Gruppe `users` die Ausführung von Snapper Aktionen.

Der Schlüssel `TIMELINE_CREATE=no` verhindert das stündlich ein Snapshot des jeweiligen Subvolumens hinzu kommt.

Wir können einzelne *Schlüssel=Wert* Paare auch auf der Kommandozeile ändern. Im Beispiel verringern wir in der Konfiguration `root` die Anzahl der gehaltenen, nummerierten Snapshots.

```
# snapper -c root set-config NUMBER_LIMIT=20
```

Jetzt bleiben die jüngsten zehn statt fünfundzwanzig Pre- und Post-Snapshot Paare nach APT Aktionen erhalten. Für den Standardgebrauch eines Laptops oder PCs dürfte dieser Wert ausreichen.

Berücksichtigt man die Snapper *default* Konfiguration mit aktivem `TIMELINE_CREATE` Schlüssel und einer APT Aktion täglich, summieren sich die Snapshots innerhalb eines Monats auf über 80. Außerdem vagabundiert der allererste *Timeline Snapshot* sage und schreibe mindestens zehn Jahre in unserem Dateisystem. Wer möchte sein System auf diesen Snapshot zurücksetzen und die ganzen Daten so lange behalten?

Man beachte: Snapper und Snapshots sind kein Mittel zur Datensicherung. Sie ermöglichen das zeitnahe Zurücksetzen des Systems bei auftretenden Fehlern, oder durch uns angestoßene Aktionen, die ungewollte Auswirkungen nach sich zogen.

An dieser Stelle sollte jeder siduction Nutzer abwägen wie viele Snapshots er wie lange halten möchte und die Konfiguration entsprechend anpassen. Die Vorlage `rolling`, die die Besonderheiten von siduction berücksichtigt, bietet hierfür einen guten Ausgangspunkt.

Unter Verwendung dieser Vorlage generiert Snapper mit dem folgenden Befehl die Konfiguration für das im Kapitel *Btrfs* erstellte Subvolumen `@data`.

```
# snapper -c data_pr create-config -t rolling /data
```

Dies:

1. Erstellt die Konfigurationsdatei `/etc/snapper/configs/data_pr` basierend auf der Vorlage `/usr/share/snapper/config-templates/rolling`.
2. Erstellt das Subvolumen `/data/.snapshots`, in dem zukünftige Snapshots von `@data` gespeichert werden. Der Pfad eines Snapshots lautet `/data/.snapshots/#/snapshot`, wobei `#` die Nummer des Snapshots ist.
3. Fügt den Namen der Konfiguration `data_pr` zum Schlüssel `"SNAPPER_CONFIGS"` in der Datei `/etc/default/snapper` hinzu.

Jetzt ist die Konfiguration aktiv. Wenn der Schlüssel `TIMELINE_CREATE=yes` gesetzt und die systemd Unit `snapper-timeline.timer` aktiv ist, übernimmt systemd die zeitgesteuerte Erstellung von Snapshots.

### 7.5.2 Snapper und systemd

Snapper installiert drei systemd Unit Paare, um in Abhängigkeit von APT Aktionen und Zeit Snapshots zu erstellen oder zu löschen.

- Beim Erstellen von Snapshots mit den Schlüsseln `DISABLE_APT_SNAPSHOT="no"` in der Datei `/etc/default/snapper` unter Mitwirkung der Systemd Unit `grub-btrfs.path` und `grub-btrfs.service` und `TIMELINE_CREATE="yes"` in den Konfigurationsdateien der Subvolumen unter Mitwirkung der Systemd Unit `snapper-timeline.timer` und `snapper-timeline.service`.
- Beim Löschen von Snapshots mit den Schlüsseln `EMPTY_PRE_POST_CLEANUP=yes`, `NUMBER_CLEANUP=yes`, `TIMELINE_CLEANUP=yes` in den Konfigurationsdateien der Subvolumen

unter Mitwirkung der Systemd Unit

`snapper-cleanup.timer` und `snapper-cleanup.service`.

Dass Snapper zu jeder APT Aktion einen Pre- und Post-Snapshot erstellt, sollte man in siduction auf jeden Fall beibehalten. siduction ist ein Rolling Release basierend auf Debian sid. Es ist durchaus möglich bei einem Upgrade einzelne, nicht wie vorgesehen funktionierende Pakete zu erhalten. Ein Rollback mit Snapper ist dann für den Benutzer eine gute Alternative, um weiterhin zuverlässig zu arbeiten.

Dagegen bietet die *TIMELINE* Funktion Raum für individuelle Anpassungen. Die richtigen Adressaten sind die beiden Timer-Units `snapper-timeline.timer` und `snapper-cleanup.timer`. Erstere ist der Zeitgeber für die Erstellung von Snapshots, die zweite bestimmt den Zeitpunkt des Entfernens von alten und leeren Snapshots.

Die Handbuchseite [systemd-timer](#) erklärt die Funktionsweise der Timer Unit.

Jetzt wenden wir uns dem Inhalt der systemd Unit `snapper-timeline.timer` im Verzeichnis `/usr/lib/systemd/system/` zu.

```
[Unit]
Description=Timeline of Snapper Snapshots
Documentation=man:snapper(8) man:snapper-configs(5)

[Timer]
OnCalendar=hourly

[Install]
WantedBy=timers.target
```

Mit dem Befehl

`systemctl edit --full snapper-timeline.timer`

öffnen wir einen Texteditor und ändern die Sektion **[Timer]** zu

`OnCalendar=*-*-* 00/02:00:00`

Snapper erstellt fortan ausgehend von Mitternacht alle zwei Stunden timeline Snapshots. Die Zeit- und Datumsspezifikationen bitte unter `man systemd.time` nachlesen.

Wir speichern die Datei und schließen den Editor. systemd legt die geänderte

Datei mit gleichem Namen im Verzeichnis `/etc/systemd/system/` an und führt den Befehl `systemctl daemon-reload` aus, um die geänderte Konfiguration zu laden.

Die zweite systemd Timer Unit `snapper-cleanup.timer` kümmert sich um die Entsorgung alter, überzähliger und leerer Snapshots. Sie hat folgenden Inhalt:

```
[Unit]
Description=Daily Cleanup of Snapper Snapshots
Documentation=man:snapper(8) man:snapper-configs(5)

[Timer]
OnBootSec=10m
OnUnitActiveSec=1d

[Install]
WantedBy=timers.target
```

Mit dem Wissen um den Inhalt des TIMELINE-Timers können wir nun abwägen, ob die Konfiguration sinnvoll ist. Für jemanden, der seinen PC jeden Tag neu startet, dürfte der Schlüssel `OnBootSec=10m` eher ungünstig sein, wenn er feststellt, dass sich am Vortag kurz vor Feierabend ein gravierender Fehler eingeschlichen hat. Sinnvoller ist für diesen Fall vermutlich den Schlüssel auf `OnBootSec=3h` einzustellen. Die Änderung der Datei erfolgt analog dem zuvor gezeigten Beispiel.

### 7.5.3 Snapper - manuelle Snapshots

Selbstverständlich können wir mit Snapper auch unabhängig von den automatischen Aktionen Snapshots erstellen. Dafür muss der ausführende Benutzer in der Snapper Konfiguration des Subvolumens mit Gruppen- oder Userrechten eingetragen sein.

Die Syntax des Befehls entspricht dem folgenden Muster, das auch die häufig zur Anwendung kommenden Optionen zeigt.

```
# snapper -c <config_name> create -t <type> -d <description>
-c <cleanup-algorithm> -u <userdata>
```

- **snapper -c <config\_name> create**

Der Snapper Befehl erstellt von dem Subvolumen der benannten Konfiguration einen Snapshot. Fehlt die Option, so wendet Snapper den Befehl auf das Btrfs default Subvolumen (das Wurzelverzeichnis */*) mit der Konfiguration **root** an. Diese Regel gilt für alle Snapper Befehle.

- **-t <type>**

Die Art des zu erstellenden Snapshots. Mögliche Werte: **single**, **pre**, **post**.

- **-d <description>**

Beliebiger Text. Bei enthaltenen Leer- und Sonderzeichen **"** verwenden.

- **-c <cleanup-algorithm>**

Die Option bestimmt, nach welchen Regeln der Snapshot automatisch gelöscht werden soll. Mögliche Werte: **number**, **timeline**, **pre**, **post**. Fehlt diese Option, bleibt der Snapshot so lange erhalten bis der Benutzer ihn manuell löscht.

- **-u <userdata>**

Legt Benutzerdaten für den Snapshot fest. Das Format muss *Schlüssel=Wert* sein. Mehrere Benutzerdaten müssen durch ein Komma getrennt sein. Zum Beispiel **author=Tom,important=yes**.

Snapper legt die Snapshots grundsätzlich im *read-only* Modus an. Man kann die Voreinstellung mit der Option **--read-write** ändern. Eine Änderung von Daten in einem Snapshot führt zu inkonsistenten Datenbeständen. Wir raten dringend davon ab, es sei denn man weiß genau, was man tut und warum.

Nun legen wir einen Snapshot an und lassen uns die Snapshots der gleichen Konfiguration anzeigen. (Die Spalten wurden gekürzt.)

```
$ snapper -c data_pr create -t single -d "AB finished" -c ↵
  number -u user=Tom
$ snapper -c data_pr list
#|Typ   |Pre #|Date   |User|Cleanup|Description|Userdata
--+-+---+-----+-----+-----+-----+-----
0|single|    |      |root|      |current  |
```

```
88|single|      |22:00:38|root|timeline|timeline  |
90|single|      |11:34:41|root|timeline|timeline  |
91|single|      |11:36:23|user|number  |AB finished|user=Tom
```

Der von uns (user) erstellte Snapshot hat die # 91. Leider ist uns der Fehler unterlaufen das der Snapshot nach der Cleanup Regel *number* behandelt wird. Das ändern wir mit der Option *modify -c ""* damit Snapper ihn nicht automatisch löscht.

```
$ snapper -c data_pr modify -c "" 91
$ snapper -c data_pr list
#|Typ    |Pre #|Date      |User|Cleanup |Description|Userdata
--+-+-----+-----+-----+-----+-----+-----+-----
0|single|    |      |root|      |current    |
88|single|    |22:00:38|root|timeline|timeline  |
90|single|    |11:34:41|root|timeline|timeline  |
91|single|    |11:36:23|user|      |AB finished|user=Tom
```

Der Snapshot # 91 bleibt jetzt so lange erhalten bis wir ihn selbst löschen.

### Snapshot löschen

Wir können zu jeder Zeit einen beliebigen Snapshot löschen, sofern wir die Rechte dazu haben. Für Snapper ist die Löschaktion nicht von Belang, denn der Cleanup Algorithmus prüft bei jedem Durchlauf neu, welche Snapshots gehalten werden. Das obere Kapitel [Snapper Konfiguration](#) erklärt darüber hinaus ausführlich die Einstellungen mit denen wir den Cleanup Algorithmus bei Bedarf anpassen.

Der folgende Befehl entfernt den Snapshot # 91 aus dem Subvolumen *@data*.

```
$ snapper -c data_pr delete 91
```

Der Befehl *delete 34-50* löscht eine Reihe von Snapshots.

Der Snapshot # 0 mit der Beschreibung *“current”* ist nicht löscher. Es ist der Snapshot, der im Dateibaum eingehangen ist und in dem wir zur Zeit arbeiten.

### 7.5.4 Snapper Rollback

Sollte einmal durch eine von uns angestoßene, völlig aus dem Ruder gelaufene Aktion, oder durch ein fehlerhaftes Upgrade das System beschädigt sein, ermöglicht Snapper mit dem *“Rollback”* das System in einen oder mehrere Zustände zurück zu versetzen, der vor dem Auftreten der Probleme vorlag.

#### Voraussetzungen

Ein *“Rollback”* wird nur mit Btrfs für das root Dateisystem unterstützt. Das root Dateisystem muss sich auf einem einzelnen Gerät, in einer einzelnen Partition und auf einem einzelnen Subvolumen befinden. Verzeichnisse, die aus / Snapshots ausgeschlossen sind, beispielsweise /root, können sich auf separaten Partitionen befinden.

#### Rollback durchführen

Vor dem Rollback testen wir erst einmal, ob das Rollbackziel unseren Erwartungen entspricht. Dazu booten wir unter Verwendung des Submenüs *siduction snapshots* in den gewünschten Snapshot, zum Beispiel 13. Das System bootet im *read-only* Modus. Die Fehlermeldung zu *sddm* ignorieren wir.

Arbeitet das System wie erwartet, kehren wir mit einem Reboot in das derzeitige default Subvolumen zurück. Dort führen wir den Rollback als **root** aus:

```
# snapper --ambit classic rollback 13
Anwendungsbereich ist classic
Nur-Lesen-Schnappschuss des Standard-Subvolumens erstellen. ↵
(Schnappschuss 15.)
Lesen-Schreiben-Schnappschuss des Schnappschusses 13 ↵
erstellen. (Schnappschuss 16.)
Einstellung des Standard-Subvolumens zu Schnappschuss 16.
```

#### Rollback immer aus dem default Subvolumen mit Angabe der Subvolumen Nummer des Rollbackziels ausführen.

Die Ausgabe beschreibt präzise den Ablauf des Rollbacks. Anschließend wird automatisch die Menüdatei *grub.cfg* des Bootmanagers Grub aktualisiert. Der Standard Booteintrag bootet jetzt in das neue Standard-Subvolumen (Schnappschuss 16). Zusätzlich erscheinen die neuen Snapshots im Submenü *siduction snapshots*. Bei Verwendung des Bootmanagers *systemd-boot* werden für alle

im Snapshot 16 enthaltenen Kernel Booteinträge erstellt. Auch hier wird der Snapshot 16 zum Standard Bootziel.

Der Befehl **snapper list** zeigt, dass wir uns unmittelbar nach dem Rollback noch in Snapshot 12 befinden und Snapshot 16 das neue Standard-Subvolumen ist. (Das Minus - hinter #12 und das Plus + hinter #16.)

```
# |Typ  |Pre #|Date    |User |Cleanup| Description  |
---+-----+-----+-----+-----+-----+
0 |single|    |        |root |        |current       |
12-|single|    |17:28:15|root |number |important     |
13 |pre   |    |11:34:41|root |number |apt           |
14 |post  |  13|11:35:56|root |number |apt           |
15 |single|    |12:05:23|root |number |rollback backup|
16+|single|    |12:05:23|root |        |r/w copy of #13|
```

Nach einem Reboot zeigt der \* hinter #16 an, dass wir uns in diesem Snapshot befinden und er das Standard-Subvolumen ist.

```
# |Typ  |Pre #|Date    |User |Cleanup| Description  |
16*|single|    |12:05:23|root |        |r/w copy of #13|
```

### 7.5.5 Datei Rollback im Root-Dateisystem

Es handelt sich dabei um das Rückgängigmachen von Änderungen an Dateien. Zu diesem Zweck werden zwei Snapshots miteinander verglichen und dann die gewünschte geänderte Datei herausgesucht. Anschließend lässt man sich die Änderungen anzeigen und entscheidet, ob sie zurückgenommen werden sollen.

Die Ausgabe von **snapper list** zeigt die aktuell vorhandenen Snapshots des Wurzeldateisystems. (Die Spalten wurden gekürzt.) Alle Snapshots mit einer Ziffer # größer Null bilden den Zustand des Dateisystems zu exakt diesem Zeitpunkt ab. Die einzige Ausnahme ist der mit einem \* gekennzeichnete. In ihn wurde gebootet und er ist der Standard-Snapshot. Wurde noch kein System Rollback vorgenommen, tritt Snapshot 0 an seine Stelle.

```
# |Typ  |Pre #|Date    |User |Cleanup| Description|Us..
---+-----+-----+-----+-----+-----+-----+
0 |single|    |        |root |        |current    |
```

```

42 |single|      |09:50:36|root |      |IP pc1      |
43 |pre   |      |11:30:18|root |number |apt        |
44 |post  |      43|11:34:41|root |number |apt        |
45*|single|      |22:00:38|root |      |            |
46 |single|      |23:00:23|root |timeline|timeline  |

```

Der Vergleich zwischen zwei Snapshots erfolgt mit:

```

# snapper status 42..45
[...]
c..... /etc/group
+..... /etc/group-
c..... /etc/hosts
[...]

```

Jede Zeile benennt eine Datei und die Art der Änderung. Ein **+** am Anfang der Zeile bedeutet, dass die Datei erstellt, ein **-**, dass die Datei gelöscht und ein **c**, dass der Inhalt der Datei geändert wurde.

Umfasst die Ausgabe sehr viele Zeilen, leiten wir sie mit der Option **-o </Pfad> /Name>** in eine Datei um.

Die Anzeige der Unterschiede einer Datei zwischen zwei Snapshots erfolgt mit:

```

# snapper diff 42..45 /etc/hosts
--- /.snapshots/42/snapshot/etc/hosts
+++ /.snapshots/45/snapshot/etc/hosts
@@ -5,5 +5,3 @@
    ff02::2    ip6-allrouters
    # This host address
    127.0.1.1  lap1
-# added 2022-12-02
-192.168.3.1  pc1

```

Wollen wir die Änderung rückgängig machen, benutzen wir den Befehl:

```

# snapper undochange 42..45 /etc/hosts

```

Ein *“Datei Rollback”* innerhalb des root Dateisystems ergibt nur dann Sinn, wenn ein Snapshot für ein *“System Rollback”* vorbereitet werden soll, oder der

Snapshot beteiligt ist, in den das System gebootet wurde (erkennbar an der Markierung \*). Eventuell ist danach der Neustart von Services oder Daemon, oder sogar ein Reboot notwendig.

Man darf dem Befehl auch mehrere Dateien getrennt durch Leerzeichen mitgeben.

#### Vorsicht

Wird der Befehl **snapper undochange 42..45** ohne die Angabe einer Datei abgesetzt, macht Snapper alle Änderungen zwischen den Snapshots 42 und 45 rückgängig. Die bessere Variante für ein solches Vorhaben ist ein *“System Rollback”*.

### 7.5.6 Datei Rollback von User Daten

#### Mit Snapper allein

Snapper behandelt den Snapshot 0 zwar wie einen Snapshot, aber er stellt den aktuellen Zustand des Subvolumens dar und ist damit variabel. Alle anderen Snapshots bilden, wie bereits zuvor erwähnt, den Zustand des Dateisystems zu exakt diesem Zeitpunkt ab. Änderungen zwischen diesen Snapshots agieren demnach nur in der Vergangenheit.

Für uns bedeutet das, dass ein *“Datei Rollback”* von User Daten zwischen den Snapshots 15 und 17 wertlos ist, da der Vorgang den aktuellen Zustand in unserem Subvolumen nicht betrifft. Wir benötigen also immer den Snapshot 0 als Ziel für Änderungen.

Wir schauen uns einen derartigen Vorgang anhand der Datei **Test.txt** im Subvolumen **@data** an.

```
$ snapper -c data_pr list
#|Typ  |Vor #|Datum  |Benutzer|Bereinigen|Beschr.
---+-----+-----+-----+-----+-----+-----
0 |single|    |      |root   |          |current
15|single|  |12:50:48|root   |timeline |timeline
16|single|  |13:51:08|root   |timeline |timeline
17|single|  |14:51:26|root   |timeline |timeline
```

Der Vergleich zwischen den Schnapshots 15 und 16:

```
$ snapper -c data_pr status 15..16
```

```
[...]  
+..... /data/user1/Test.txt  
[...]
```

Die Datei erscheint erstmals im Snapshot 16. Wir vergleichen mit dem nächsten Snapshot.

```
$ snapper -c data_pr status 16..17  
[...]  
c..... /data/user1/Test.txt  
[...]
```

Die Datei wurde zwischen den Snapshots 16 und 17 verändert.  
Es folgt eine Abfrage mit `diff`, die die Änderungen zwischen 16 und 17 ausgibt.

```
$ snapper -c data_pr diff 16..17 /data/user1/Test.txt  
--- /data/.snapshots/16/snapshot/user1/Test.txt  
+++ /data/.snapshots/17/snapshot/user1/Test.txt  
@@ -8,6 +8,8 @@  
  Testdatei  
  
  Dieser Text stand schon vor  
  dem Snapshot 16 in der Datei.  
  
-Dieser auch, aber er wurde gelöscht.  
+  
+Dieser Text wurde nach dem  
+Snapshot 16 eingefügt.
```

Da die Datei seit dem Snapshot 17 nicht mehr verändert wurde, erzeugt der Befehl

```
$ snapper -c data_pr diff 16..0 /data/user1/Test.txt
```

für den Vergleich von Snapshot 16 mit dem aktuellen Inhalt der Datei keine andere Ausgabe.

Nun setzen wir den `undochange` Befehl zwischen 16 und 0 ab. Danach enthält die `Test.txt` die ersten sechs Zeilen aus dem Snapshot 16.

```
$ snapper -c data_pr undochange 16..0 /data/user1/Test.txt
```

```
angelegt:0 geändert:1 gelöscht:0
```

```
$ cat /data/user1/Test.txt  
Testdatei
```

Dieser Text stand schon vor  
dem Snapshot 16 in der Datei.

Dieser auch, aber er wurde gelöscht.

Eine gelöschte Datei wird mit dem gleichen Befehl zurück in das aktuelle Verzeichnis befördert. Nur die Rückmeldung von Snapper ändert sich geringfügig.

```
$ snapper -c data_pr undochange 16..0 /data/user1/Test.txt  
angelegt:1 geändert:0 gelöscht:0
```

### Mit Snapper und Meld

Die vorangegangene Vorgehensweise stellt immer eine Datei als ganzes auf den Stand zurück, der dem ausgewählten Snapshot entspricht. Einzelne Teile der Änderungen können wir so nicht übernehmen.

Das Vergleichsprogramm **Meld** füllt genau diese Lücke. *Meld* ist zusätzlich in der Lage per *Copy & Paste* Teile an beliebiger Stelle im aktuellen Dokument einzufügen. Ein Vorteil auch gegenüber **Kompare** des KDE Desktops. In siduction wird *Meld* nicht standardmäßig installiert. Wir holen dies nach.

Die Aktionen von Snapper sind für den nicht **root** Benutzer immer dann möglich, wenn in der Konfigurationsdatei für das Subvolumen der Schlüssel `ALLOW_GROUPS=users` eingestellt ist. Dies ist Standard. Jedoch bleibt ihm der Zugriff auf die Dateien des Snapshots innerhalb des Dateisystems verwehrt, weil das Verzeichnis `/.snapshots` nur für **root** lesbar und ausführbar ist. Um mit *Meld* arbeiten zu können ändern wir dies.

Snapshots für Benutzer lesbar machen und *Meld* installieren. (Ausführen als **root**.)

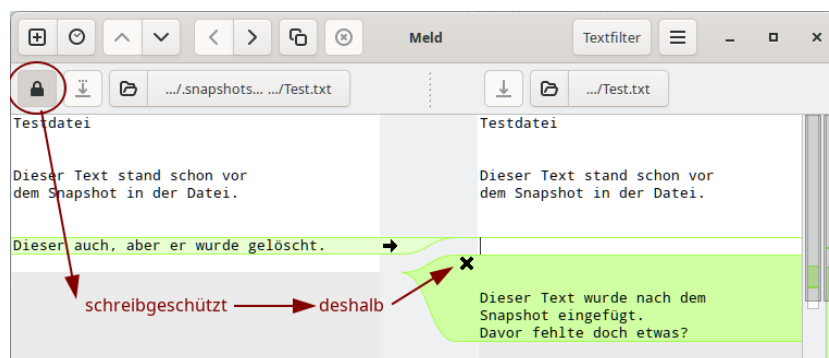
```
# chmod a+rx /data/.snapshots  
# apt update && apt install meld
```

Zur Erinnerung: Snapshots in Btrfs sind immer im nur-les-Modus gespeichert. Einzige Ausnahme ist der **System Rollback** Snapshot.

Wir benutzen zur Auswahl der Dateiänderungen Snapper in der gleichen Art wie zuvor. Der Befehl `$ snapper -c data_pr diff 16..0 /data/user1/Test.txt` enthält den genauen Pfad zur Datei `Test.txt` in dem Snapshot.

```
$ snapper -c data_pr diff 16..0 /data/user1/Test.txt
--- /data/.snapshots/16/snapshot/user1/Test.txt
+++ /data/user1/Test.txt
[...]
```

Wir starten *Meld* und wählen für den Dateivergleich die beiden Dateien mit den Pfaden aus. Die Unterschiede sind sofort sichtbar.



Ein Klick auf den Pfeil überträgt die Zeile in unsere aktuelle Datei. Ein weiterer Klick auf das Kreuz entfernt die anderen Zeilen. Eine Übertragung auf die Datei im Snapshot ist nicht möglich, da das Dateisystem des Snapshots schreibgeschützt ist.

Da uns Snapper den genauen Pfad zu unserer Datei im Snapshot anzeigt, haben wir auch die ganz konventionelle Möglichkeit eine Datei aus dem Snapshot in unser aktuelles Arbeitsverzeichnis zu kopieren.

```
$ cp /data/.snapshots/16/snapshot/user1/Test.txt /home/user1/
/Test.txt
```

### 7.5.7 Quellen BTRFS und Snapper

- `man btrfs` und `man btrfs-subvolume` sowie weitere Unterseiten von `"man btrfs"`.

- [Btrfs im wiki.debianforum.de](https://wiki.debianforum.de)
- [Btrfs wiki von kernel.org](https://wiki.kernel.org)
- [Btrfs Dokumentation](#)
- [Btrfs Snapshot im grub Menü](#)
- `man snapper` und `man snapper-configs`
- [Snapper Projektseite](#)
- [Snapper auf GitHub](#)
- `man systemd.time` und `man systemd.timer`

Zuletzt bearbeitet: 2024-12-17

## 7.6 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](#)

### 7.6.1 sources.list.d - Liste der Quellen

Das APT-System benötigt mindestens eine Konfigurationsdatei, welche Informationen über den Ort der installierbaren und aktualisierbaren Pakete beinhaltet. Im allgemeinen nennt man diese Datei `<sourcename>.sources`. siduction stellt die Quellen in dem Ordner `/etc/apt/sources.list.d/` bereit. Innerhalb dieses Verzeichnisses befinden sich standardmäßig folgende Dateien:

```
debian.sources
extra.sources
fixes.sources
```

Die Aufteilung in mehrere Dateien erleichtert die Auswahl von Spiegelserversn ("mirror switching") und das Ergänzen oder Austauschen von Quellen-Listen.

Eigene Quellen-Listen können mit der Benennung `/etc/apt/sources.list.d/<sourcename>.sources` hinzugefügt werden.

Im Januar 2025 führte Debian im Unstable Zweig das Format *deb822* für die Quellen-Listen ein. Die bisher einzeiligen Einträge ersetzt jetzt eine Reihe von aufeinander folgenden Schlüssel-Wert Paaren. Eine Leerzeile trennt die Quellen voneinander. Neu ist der für jede Quelle erforderliche Schlüssel *Signed-By*. Die Aktivierung und Deaktivierung erfolgt über den Schlüssel *Enabled* und ersetzt das zuvor verwendete Kommentarzeichen.

Damit der Umstieg zu dem aktuellen Format leichter gelingt, liefert apt ein Transformationsskript mit. Der Aufruf benötigt **root** Rechte und erfolgt mit `apt &#x2197; modernize-sources`.

Die Quellen-Liste `/etc/apt/sources.list.d/extra.sources` sieht bei siduction im Format *deb822* zum Beispiel so aus:

```
# This is the default mirror, choosen at first boot.
```

```
# One might consider to choose the geographical nearest
# or the fastest mirror.
Types:      deb
URIs:       https://packages.siduction.org/extra
Suites:     unstable
Components: main contrib non-free non-free-firmware
Enabled:    yes
Signed-By:  /usr/share/keyrings/siduction-archive-keyring.gpg
            gpg
[...]
```

und `/etc/apt/sources.list.d/debian.sources` enthält dann das eigentliche Debian Repo:

```
# debian loadbalancer
Types:      deb
URIs:       https://deb.debian.org/debian/
Suites:     unstable
Components: main contrib non-free non-free-firmware
Enabled:    yes
Signed-By:  /usr/share/keyrings/debian-archive-keyring.gpg
[...]
```

Weitere Einträge für optionale siduction Repositories finden sich auf [siduction Repositories](#).

Fügt man zum Beispiel ein oder mehrere Debian Repositories hinzu, so würde dies folgendermaßen aussehen:

```
# Debian
## Unstable
Types:      deb
URIs:       http://ftp.us.debian.org/debian/
Suites:     unstable
Components: main contrib non-free non-free-firmware
Enabled:    yes
Signed-By:  /usr/share/keyrings/debian-archive-keyring.gpg

## Testing
```

```

Types:      deb
URIs:       http://ftp.us.debian.org/debian/
Suites:     testing
Components: main contrib non-free non-free-firmware
Enabled:    no
Signed-By:  /usr/share/keyrings/debian-archive-keyring.gpg

## Experimental
Types:      deb
URIs:       http://ftp.us.debian.org/debian/
Suites:     experimental
Components: main contrib non-free non-free-firmware
Enabled:    no
Signed-By:  /usr/share/keyrings/debian-archive-keyring.gpg

```

**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.](#)

**7.6.2 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
<a href="#">apt update</a>	apt-get update	Auffrischen der Paketdatenbank.

apt	apt-get	Kurzinfo
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 full-upgrade -d	apt-get dist-upgrade -d	Aktualisierung des 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 des Paketes installiert.
-	apt-mark unhold	Hebt den Befehl 'apt-mark hold' auf.
apt search	apt-cache search	Sucht entsprechend des eingegebenen Musters nach Paketen. (regex möglich)

apt	apt-get	Kurzinfo
<a href="#">apt show</a>	apt-cache show	Anzeige der Details eines Paketes.
apt policy	apt-cache policy	Zeigt die installierte, oder installierbare Version eines Paketes.

### 7.6.3 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 garantieren, 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.
```

### 7.6.4 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+F3** auf die Textkonsole wechseln.
3. Einloggen als **root**.

Anschließend das gewünschte Programmpaket installieren:

```
systemctl isolate multi-user.target
apt update
apt install <Paketname>
systemctl isolate graphical.target && 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 ...
```

### 7.6.5 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.

### 7.6.6 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:

Wir editieren die Datei `/etc/apt/sources.list.d/debian.sources` mit **root** Rechten. Bei den Quellen von Unstable erhält der Schlüssel *Enabled* den Wert *no* und bei Testing setzen wir den Wert *yes*. 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.sources` wieder rückgängig. Nach dem Speichern der Änderungen:

```
apt update
```

Wenn ein neues, fehlerfreies Paket in sid eintrifft, kann man die neueste Version wieder installieren, nachdem man den "hold"-Status beendet hat:

```
apt-mark unhold kmahjongg
apt update
apt install kmahjongg
    oder
apt full-upgrade
```

### 7.6.7 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 Spiegelservers 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-Repositories stammen oder selbst kompi-

liert 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.

#### 7.6.8 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 einzelnen Pakets (hier z. B. xterm 397-1) erfolgt mit:

```
root@siduction# apt install xterm=397-1
Upgrading:
  xterm

Summary:
  Upgrading: 1, Installing: 0, Removing: 0, Not Upgrading: 12
  Download size: 860 kB
  Space needed: 1.024 B / 176 GB available

Holen:1 https://deb.debian.org/debian unstable/main amd64
      xterm amd64 397-1 [860 kB]
```

```
Es wurden 860 kB in 0 s geholt (1.760 kB/s).
Changelogs werden gelesen... Fertig
(Lese Datenbank ... 456325 Dateien und Verzeichnisse sind
  derzeit installiert.)
Vorbereitung zum Entpacken von .../archives/xterm_397-1
  _amd64.deb ...
Entpacken von xterm (397-1) über (396-1) ...
xterm (397-1) wird eingerichtet ...
```

### (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 im multi-user.target 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 bsutils 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.

### 7.6.9 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+F3** auf die Textkonsole wechseln.
3. Einloggen als **root**.

Anschließend folgende Befehle ausführen:

```
systemctl isolate multi-user.target
apt update
apt full-upgrade
apt clean
systemctl isolate graphical.target && exit
```

Wurde ein neuer Kernel installiert, ist an Stelle von `“systemctl isolate graphical.target”` der Befehl `systemctl reboot` notwendig, um in den neuen Kernel zu booten.

#### 7.6.10 Warum ausschließlich apt verwenden

Zum Installieren, Löschen und Durchführen einer Systemaktualisierung soll `apt` verwendet werden.

Bitte von Systemaktualisierungen mit Anwendungen wie `synaptic` oder `discover` 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 `synaptic` und `discover` 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.

### 7.6.11 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.

#### 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 *khangman* und *logmanager*.

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 den aktiven Quellen) 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".

```

Aktionen  Rückgängig  Paket  Auflöser  Suchen  Optionen  Ansichten  Hilfe
C-T: Menu  ?: Help  q: Quit  u: Update  g: Preview/Download/Install/Remove Pkgs
aptitude 0.8.13 @ pcl
i A  git-man 1:2.29.2-1 1:2.29.2-1
i  gitmagic 20160304-1.2 20160304-1.2
i  gman 0.9.3-5.3 0.9.3-5.3
i A  gnuplot-data 5.4.0+dfsg1-1 5.4.0+dfsg1-1
i A  info 6.7.0.dfsg.2-5 6.7.0.dfsg.2-5
i A  install-info 6.7.0.dfsg.2-5 6.7.0.dfsg.2-5
i A  kdoctools5 5.74.0-2 5.74.0-2
i A  khelpcenter 4:20.04.2-1 4:20.04.2-1
i  libreoffice-help-de 1:7.0.3-4 1:7.0.3-4
i  man-db 2.9.3-2 2.9.3-2
i  man2html 1.6g-12 1.6g-12
i A  man2html-base 1.6g-12 1.6g-12
i  mandoc 1.14.4-1 1.14.4-1
i  manpages 5.09-2 5.09-2
i  manpages-de 4.1.0-1 4.1.0-1

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.

The default manual page viewer is a terminal window with the original man(1). It can also
launch gv, evince, or a link to a CGI script which utilizes man2html, for viewing manual
pages using a web browser.

There is an index search function to look for the man pages that one needs. It's simple, but
it's useful.
Markierungen: 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

```

## 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

```

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: 2025-10-04

## 7.7 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 Spiegelservers geladen werden, gleichgültig wie viele Geräte diese Pakete benötigen. Dies spart Netzwerkbandbreite, erhöht die Geschwindigkeit für die Benutzer und reduziert die Last an den Spiegelservers.

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.

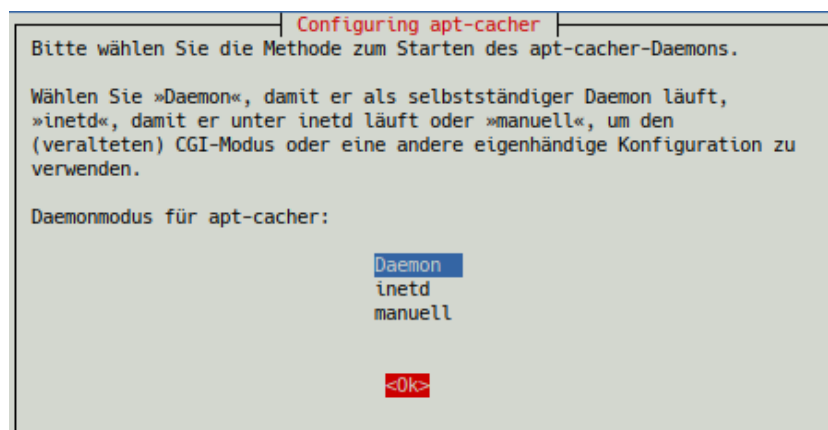
#### 7.7.1 Server installieren

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 libfileysys-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.



Der empfohlene Deamonmodus *“daemon”* 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.

### 7.7.2 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.`  
`sources`, `extra.sources` und `fixes.sources`.

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

Zuletzt bearbeitet: 2026-01-02

## 7.8 Nala Paketverwaltung

### Benutzerfreundlicher und leistungsstärker als APT

Nala ist ein Kommandozeilen-Frontend für den APT-Paketmanager. Es benutzt die `python-apt` API statt der APT-Bibliotheken zur Verwaltung der Pakete. Das Ziel von Nala ist eine übersichtlichere und benutzerfreundlichere Darstellung sowohl des aktuellen Paketbestands als auch der angeforderten Aktionen und deren Durchführung. Darüber hinaus soll der Download von Paketen beschleunigt werden.

Nala verwendet viele gleichlautende Befehle von APT, wie etwa `install`, `remove`, `purge`, `update`, `show` und `search`. Außerdem implementiert es den Befehl `history`, um vergangene Transaktionen zu sehen und dem Benutzer die Möglichkeit zu geben diese rückgängig zu machen, und den Befehl `fetch`, der eine Liste der schnellsten Spiegelserver zur Auswahl anzeigt. In der Grundeinstellung beschleunigt Nala den Download dadurch, dass drei Pakete gleichzeitig von einem Server geholt werden. Das Limit von drei Verbindungen pro Spiegelserver besteht, um die Belastung der Spiegelserver zu minimieren.

#### 7.8.1 Nala verwenden

Ab siduction 2022.1.0 wird Nala automatisch installiert und ist sofort verwendbar. Es kann jederzeit ohne Weiteres zwischen APT und Nala gewechselt werden. Ein Blick in die manpage `man nala` sollte obligatorisch sein. Vor der Anwendung empfehlen wir dringend zwei Änderungen in der Konfigurationsdatei `/etc/nala/nala.conf` vorzunehmen.

Zum einen den Wert für die Konfigurationsoption `auto_remove`, diesen ändern wir zu `false`, so wie es das folgende Listing zeigt:

```
# Set to false to disable auto auto-removing
auto_remove = false
```

Zum anderen den Wert `full_upgrade = false`, dieser muss auf `true` geändert werden, damit automatisch ein `full-upgrade` ausgeführt wird.

```
# Set to true to make full-upgrade the default
full_upgrade = true
```

Der Grund für beide Änderungen ist, dass siduction auf *GNU Linux debian unstable/sid* basiert und diese Basis sich täglich verändert und deshalb regelmäßig auf den neusten Stand gebracht werden muss. Bei einem Upgrade von *debian sid* kann gelegentlich eine Situation entstehen, in der wesentliche Teile des Systems entfernt werden sollen. Mit der Option `auto_remove = true` haben wir keine Möglichkeit zu recherchieren, zu prüfen und selbst zu entscheiden, ob oder welche Pakete zu entfernen sind. Auch im normalen Betrieb sollten Pakete nicht mit `auto_remove`, sondern erst nach einer Sichtkontrolle entfernt werden.

Ist es im Ausnahmefall nötig auf einen `full-upgrade` verzichten zu müssen und stattdessen nur einen `upgrade` vorzunehmen, so kann `nala` mit folgendem Befehl aufgerufen werden, `nala upgrade --no-full`.

### 7.8.2 Befehle analog zu APT

Viele der von APT bekannten Befehle sind in Nala identisch. In der Grundeinstellung erwartet Nala vor dem Ausführen einer angeforderten Aktion, die das System ändert, immer eine Bestätigung.

- **`nala update`**  
Aktualisiert die Paketinformationen der konfigurierten Paketquellen.
- **`nala install <Paketname>`**  
Installiert das benannte Paket in unser System.
- **`nala remove <Paketname>`**  
Entfernt das benannte Paket aus unserem System.
- **`nala purge <Paketname>` oder `nala remove --purge <Paketname>`**  
Entfernt das benannte Paket mit seinen Konfigurationsdateien aus unserem System.
- **`nala upgrade`**  
Führt `update`, gefolgt von `upgrade` aus. (Standardeinstellung in der Datei `/etc/nala/nala.conf`)  
Führt `update`, gefolgt von `upgrade --full` aus. (Mit den oben genannten Änderungen in der Datei `/etc/nala/nala.conf`)

Die benutzerfreundliche Aufbereitung der Ausgabe im Terminal erleichtert die Übersicht, wie das Beispiel zeigt.

(Um root-Rechte zu erlangen, wurde im Befehl “doas” verwendet.)

```

user1@pc1:~$ doas nala install yapf3
=====
Installieren
=====
Paket:          Version:        Größe:
python3-yapf      0.32.0-1          133 KB
yapf3             0.32.0-1          30 KB
=====

Zusammenfassung
=====
Installieren 2 Pakete

Speicherplatz erforderlich  892 KB

Möchtest du fortfahren? [J/n] j
- Pakete installieren -
Auspacken:      python3-yapf (0.32.0-1)
Auspacken:      yapf3 (0.32.0-1)
Konfigurieren:  python3-yapf (0.32.0-1)
Konfigurieren:  yapf3 (0.32.0-1)
Wird bearbeitet: triggers for runit (2.1.2-50)
Wird bearbeitet: triggers for man-db (2.11.0-1+b1)

✓ Ausführen von dpkg ... 100.0% • 0:00:00 • 5/5

Erfolgreich beendet

```

Im ersten Teil der Ausgabe erhalten wir eine Liste der zu installierenden Pakete mit der Angabe ihrer Versionen und Größe. Nach der Bestätigung listet der zweite Teil die ausgeführten Aktionen auf.

### 7.8.3 Befehle die APT nicht enthält

#### Befehl “fetch”

Der Befehl **nala fetch**, ohne weitere Optionen ausgeführt, ermittelt automatisch die Distribution und das Release unserer Installation, sucht nach den schnellsten Spiegelservers, listet diese zur interaktiven Auswahl auf und legt nach der Auswahl eines oder mehrerer Server die Datei `/etc/apt/sources.list.d/nala-sources.list` an.

Die Option **-c, --country** beschränkt die Suche mit Hilfe des ISO-Ländercodes. Eine mehrfache Angabe der Option ist erlaubt.

Die Option `--non-free` ergänzt die Datei um contrib- und non-free-Komponenten.

Bei einem Download werden von jedem der Server bis zu drei Pakete gleichzeitig geholt.

### Befehl “history”

Der Befehl `nala history`, ohne Unterbefehl aufgerufen, zeigt eine Zusammenfassung aller mit Nala durchgeführten Aktionen. Jede Zeile entspricht einer Aktion und enthält die ID, das Kommando, den Zeitstempel, die Anzahl der geänderten Pakete und den User, der die Aktion angefordert hat. Aktionen, die mit anderen Programmen erfolgten, werden nicht erfasst.

```
user1@pc1:~$ nala history
  ID  Command                                Date and Time          Altered  Requested-By
  ---  ---                                ---
  1    upgrade bash busybox ...             2022-11-07 17:41:13 CET      72      root (0)
  2    purge libbpf0                         2022-11-08 13:41:10 CET       1      root (0)
  3    install yapf3                         2022-11-09 17:10:37 CET       2      user1 (1000)
user1@pc1:~$ _
```

Details zu einer Aktion aus der History zeigt der gleiche Befehl mit dem angehängten Unterbefehl `info <ID>`.

```
user1@pc1:~$ nala history info 3
=====
Installiert
=====
Paket:          Version:          Größe:
python3-yapf      0.32.0-1             133 KB
yapf3             0.32.0-1             30 KB
=====
Zusammenfassung
=====
Installiert 2 Pakete
```

Wollen wir nun die Installation von “yapf3” mit den Abhängigkeiten, in unserem Fall “python3-yapf”, rückgängig machen, benutzen wir dafür den Unterbefehl `undo <ID>`.

(Auch hier erlangt **user1** root-Rechte durch die Verwendung von “doas”).

```
user1@pc1:~$ doas nala history undo 3
```

Paket:	Version:	Größe:
python3-yapf	0.32.0-1	849 KB
yapf3	0.32.0-1	43 KB

Zusammenfassung

Entfernen 2 Pakete

Speicherplatz freizugeben 892 KB

Möchtest du fortfahren? [J/n] j

Verlauf rückgängig machen 3

Entferne: yapf3 (0.32.0-1)

Entferne: python3-yapf (0.32.0-1)

Wird bearbeitet: triggers for runit (2.1.2-50)

Wird bearbeitet: triggers for man-db (2.11.0-1+b1)

✓ Ausführen von dpkg ... 100.0% • 0:00:00 • 5/5

Erfolgreich beendet

Im ersten Teil der Ausgabe sehen wir die zu entfernenden Pakete mit der Angabe ihrer Versionen und Größe. Nach der Bestätigung listet der zweite Teil die ausgeführten Aktionen auf.

Sollten wir es uns noch einmal anders überlegen und die Pakete doch wieder verwenden wollen, hilft der Befehl `nala history redo <ID>` weiter, um die Aktion noch einmal auszuführen. Mit dem Befehl `nala history clear <ID>` lassen sich Einträge aus der History entfernen, `nala history clear --all` entfernt alle Einträge.

In der hier beschriebenen Nala Version 0.11.1 unterstützen die Unterbefehle `undo <ID>` und `redo <ID>` derzeit nur die Aktionen Installieren oder Entfernen. In einer zukünftigen Version, die dann auf der Programmiersprache Rust basieren wird, sollen sich komplette Dist-Upgrades zurückrollen lassen.

Zuletzt bearbeitet: 2023-10-15

## 7.9 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.

### 7.9.1 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.

### 7.9.2 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.

### 7.9.3 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
```

Zuletzt bearbeitet: 2021-11-29

## 7.10 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.

### 7.10.1 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 [Systemd-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].

### 7.10.2 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 Einhängpunkte 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 Einhäng-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.

### 7.10.3 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.

### 7.10.4 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.

### 7.10.5 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.

### 7.10.6 Quellen systemd

[Deutsche Manpage 'systemd'](#)

[Deutsche Manpage 'systemd.unit'](#)

[Deutsche Manpage 'systemd.syntax'](#)

Zuletzt bearbeitet: 2024-08-30

## 7.11 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.

### 7.11.1 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.

### 7.11.2 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.

### 7.11.3 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.

**7.11.3.1 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.

**7.11.3.2 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 Einhängpunkt
- [\[Automount\]](#) konfiguriert einen Selbsteinhängpunkt
- [\[Swap\]](#) 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.

**7.11.3.3 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** 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>**.

#### 7.11.4 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.

```
(in einer Zeile eingeben)
$ 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.
  - 2.1. *sockets.target*  
von *basic.target* angefordert wird,
    - 2.1.1. *cups.socket*  
und alle weiteren `.socket`-Units von *sockets.target* hereingeholt werden.
    - 2.2. *paths.target*  
von *basic.target* angefordert wird,
      - 2.2.1. *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. (Ge-

nau 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.

### 7.11.5 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 Verzeichnis-

sen, 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-locale✓
             d.service → /lib/systemd/system/systemd-locale✓
             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 aller 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 (Den Befehl auf einer Zeile eingeben.)

```
$ 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.

### 7.11.6 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.

Zuletzt bearbeitet: 2025-09-19

## 7.12 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.

### 7.12.1 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](#).

### 7.12.2 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=` angegeben 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 been-

det 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 Dienstprozess 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.

		on	on	on	on	on
► Restart= ►	always	success	failure	abnormal	abort	watchdog
▼ Exit-Grund ▼						
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](#)

### 7.12.3 Quellen systemd-service

[Deutsche Manpage, systemd.service](#)

[LinuxCommunity, Systemd-Units selbst erstellen](#)

Zuletzt bearbeitet: 2021-11-29

## 7.13 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 Einhängpunkte 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 Einhängpunkt 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 Einhängpunkt, den sie steuern, benannt sein. Beispiel: Der Einhängpunkt `"/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 Einhängpunkte übersetzt systemd in der frühen Bootphase mit Hilfe des `systemd-fstab-generator`s in native mount-Units.

### 7.13.1 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`.
- **LazyUnmount=** (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 Einhängeloptionen 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.

### 7.13.2 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.

### 7.13.3 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 Webserver 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 Webserver eingehangen, sobald wir in das Verzeichnis `/home/\<user\>/www_data` wechseln. Die Statusabfrage bestätigt die Aktion.

```
# systemctl status home-<user>-www_data.mount
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 [...]
 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
home-<user>-www_data.automount Automount server1/var/www/ using 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 [...]
  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.*
[...]systemd[1]: Mounted Mount server1/var/www/ using NFS
[...]systemd[1]: Unmounting Mount server1/var/www/ using NFS
[...]systemd[1]: home-<user>-www_data.mount: Succeeded.
[...]systemd[1]: Unmounted Mount server1/var/www/ using NFS
[...]systemd[1]: home-<user>-www_data.automount: Got
                    automount request for /home/<user>/www_data
                    triggered by 2500 (Thunar)
[...]systemd[1]: Mounting Mount server1/var/www/ using NFS
[...]systemd[1]: Mounted Mount server1/var/www/ using NFS
[...]systemd[1]: Unmounting Mount server1/var/www/ using NFS
[...]systemd[1]: home-<user>-www_data.mount: Succeeded.
[...]systemd[1]: Unmounted Mount server1/var/www/ using NFS
[...]systemd[1]: home-<user>-www_data.automount: Got
                    automount request for /home/<user>/www_data
                    triggered by 6582 (bash)
[...]systemd[1]: Mounting Mount server1/var/www/ using NFS
[...]systemd[1]: Mounted Mount server1/var/www/ using NFS
[...]systemd[1]: Unmounting Mount server1/var/www/ using NFS
[...]systemd[1]: home-<user>-www_data.mount: Succeeded.
[...]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.

### 7.13.4 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](#)

Zuletzt bearbeitet: 2021-11-29

## 7.14 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”.

Mit systemd 258~rc1-1 (August 2025) wurde die Unterstützung der alten sysvinit-Befehle im laufenden System entfernt. Die Anpassung der Kernelbootzeile mit den sysvinit-Befehlen ist weiterhin möglich.

Ziel-Unit	Beschreibung
<b>emergency.target</b>	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.
<b>rescue.target</b>	Startet das Basissystem (einschließlich Systemabhängigkeiten) und eine Notfall-Shell. Im Vergleich zu multi-user.target könnte dieses Ziel als single-user.target betrachtet werden.
<b>multi-user.target</b>	Mehrbenutzersystem mit funktionierendem Netzwerk, ohne Grafikserver X. Diese Unit wird verwendet, wenn man X stoppen bzw. nicht in X booten möchte. <a href="#">Auf dieser Unit wird in besonderen Fällen (wenn X selbst oder die Desktop-Umgebung aktualisiert werden) eine Systemaktualisierung (dist-upgrade) durchgeführt.</a>
<b>graphical.target</b>	Die Unit für den Mehrbenutzermodus mit Netzwerkfähigkeit und einem laufenden X-Window-System.
<b>default.target</b>	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.

### 7.14.1 Besonderheiten

Bei den Ziel-Units sind drei Besonderheiten zu beachten:

1. Die Verwendung auf der Kernel-Befehlszeile 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-Befehlszeile 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+F3** zum virtuellen Terminal tty3 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
emergency.target	systemctl isolate emergency.target
rescue.target	systemctl isolate rescue.target
multi-user.target	systemctl isolate multi-user.target
graphical.target	systemctl isolate graphical.target

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 xxxxxx.target” aufgerufen werden sollen.

Ziel	Terminal-Befehl
halt	systemctl halt
shutdown	systemctl shutdown
reboot	systemctl reboot

“halt”, “shutdown” und “reboot” holen mehrere Units in der richtigen Reihenfolge herein, um das System geordnet zu beenden und ggf. einen Neustart auszuführen.

#### 7.14.2 Quellen systemd-target

[Manpage systemd.target, de](#)

Zuletzt bearbeitet: 2025-08-31

## 7.15 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.

### 7.15.1 Benötigte Dateien

Die **systemd-path**-Unit benötigt für ihre Funktion mindestens zwei Dateien mit vorzugsweise dem gleichen Namen, aber unterschiedlicher Namenserverweiterung 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.

### 7.15.2 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:

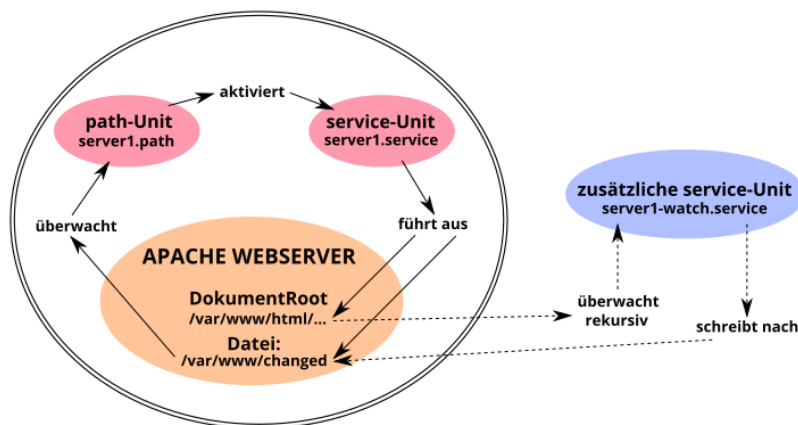
- **PathExists=**  
prüft, ob der betreffende Pfad existiert. Wenn es zutrifft, wird die zugehörige Unit aktiviert.
- **PathExistsGlob=**  
Wie oben, unterstützt Datei-Glob-Ausdrücke (siehe dazu auch die Ausgabe von `man glob`).
- **PathChanged=**  
beobachtet eine Datei oder einen Pfad und aktiviert die zugehörige Unit, wenn Änderungen auftreten.  
Aktionsauslösende Änderungen sind:
  - Erstellen und Löschen von Dateien.
  - Attribute, 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.



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.

### 7.15.3 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.

#### 7.15.4 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.

## 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.

### 7.15.5 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 [...]
   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/

[...]systemd[1]: Starting Watching server1 folder....
[...]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 [...]
  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.

### 7.15.6 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 [...]
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

[...]systemd[1]: Starting Change permissions in server1
[...]systemd[1]: server1.service: Succeeded.
[...]systemd[1]: Finished Change permissions in server1
```

### 7.15.7 Quellen systemd-path

[Deutsche Manpage 'systemd.path'](#)

Ein anders gelagertes Beispiel:

[PRO-LINUX.DE, Systemd Path Units...](#)

Zuletzt bearbeitet: 2021-11-29

## 7.16 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.

### 7.16.1 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`.

### 7.16.2 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 Beschrei-

bung 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.

### 7.16.3 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 letzt-  
malig aktiviert wurde".

Beide Optionen zusammen lösen die zugehörige service-Unit 90 Sekun-  
den 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 Ma-  
schine 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 auf-  
gerufen.

```
# systemctl start backup.service
```

### 7.16.4 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.

### 7.16.5 Quellen systemd-timer

[Deutsche Manpage ‘systemd.timer’](#)

[Archlinux Wiki, Timers](#)

[PRO-LINUX.DE, Systemd Timer Units...](#)

Zuletzt bearbeitet: 2021-11-29

## 7.17 systemd-boot

Obwohl bereits vor mehr als zehn Jahren in systemd aufgenommen, findet man den Bootmanager systemd-boot (kurz sd-boot) auf Desktop Systemen selten. Entscheidet man sich bei einem passenden Setup und nach gründlichen Tests für sd-boot, stellt der Umstieg für einen etwas geübten Nutzer keine große Herausforderung dar.

### Besonderheiten

Der Bootmanager sd-boot wurde mit dem Ziel entwickelt den Bootvorgang schnell, einfach und sicher zu gestalten. Der Anwender erhält ein textbasiertes Bootmenü. Die Anzeige kommt ohne jeglichen Schnickschnack aus.

Er ist, dem Namen entsprechend, tief in systemd integriert und greift auf dort vorhandene Dienste zu. sd-boot erstellt ein optisch minimalistisches Bootmenü. Jeder Menüeintrag beruht auf einer einzigen, eigenen Textdatei. Die Konfiguration für die Benutzerschnittstelle ist im Vergleich zu GRUB rudimentär und der Installationsumfang, mit 32 Dateien und einem zwanzigstel des Datenvolumens, sehr gering.

sd-boot ist ausschließlich für UEFI Hardware verfügbar. Dafür benötigt sd-boot zwingend eine ESP (**E**fi **S**ystem **P**artition). Empfohlen wird zusätzlich eine XBOOTLDR Partition.

Ist nur die ESP vorhanden, wird sie unter `/boot` eingehangen.

Bei ESP und XBOOTLDR Partition wird die ESP unter `/efi` und die XBOOTLDR Partition unter `/boot` eingehangen.

Dateisystemtreiber für die XBOOTLDR Partition sind ggf. unterhalb `/efi` abzuliegen.

sd-boot kann nur Betriebssysteme von Partitionen des gleichen Mediums booten. Um Betriebssysteme von weiteren Medien zu booten benutzt man die Chain-Loader Technik zu GRUB oder einem anderen Bootloader.

siduction installiert den Bootmanager GRUB automatisch. Eine Auswahl von sd-boot ist während der Installation nicht möglich. Wechselt man später zu sd-boot, so erzeugt sd-boot angepasste initrd für jeden im System vorhandenen Kernel. Nur mit diesen initrd bootet sd-boot.

### Funktionen von sd-boot:

- Von vollständig verschlüsselter Festplatte booten.

- Unterstützung für die XBOOTLDR Partition
- Laden von Drop-in-Treibern.
- Registrieren von SecureBoot-Schlüsseln.
- Erstellen eines Menüeintrages bei Installation neuer Kernel.
- Boot-Zählung

In Zusammenhang mit gescheiterten Bootvorgängen kann der Booteintrag automatisch entfernt werden.

- Unterstützung für die Übergabe eines Zufallsseeds an das OS.

Es dient dem Schutz vor der Verwendung manipulierter OS-Images.

Darüber hinaus besteht die Möglichkeit vor einem Reboot oder während des Bootvorganges Kommandos an sd-boot zu übergeben. Informationen hierzu gibt `man systemd-boot`.

### Anwendungsszenarien

Eignung bei unterschiedlichen Systemkonfigurationen und im Vergleich mit GRUB.

Konfiguration	sd-boot	GRUB	Bemerkung
Ein immutable OS	++	o	Es ist nur ein minimalistischer Bootmanager ohne Menü notwendig. GRUB ist für diesen Zweck überdimensioniert.
Ein OS auf einer HD	++	o	Bootmenü nur bei mehreren Kernen notwendig.
Mehrere Linux OS auf einer HD	+	+	sd-boot auf allen OS notwendig. GRUB benötigt externes Modul <i>os-prober</i> .
Dualboot mit WIN / MAC auf einer HD	+	+	Wie zuvor. Bei beiden booten mittels ChainLoader möglich.
Mehrere Linux OS auf mehreren HD	-	+	sd-boot kann nur OS von einer HD booten. Zwei Instanzen im UEFI mit Auswahl über die Firmware notwendig.
Dualboot mit WIN / MAC auf mehreren HD	-	+	Wie zuvor.

Konfiguration	sd-boot	GRUB	Bemerkung
Mehrere Varianten eines Linux OS auf einer HD	o	o	Bei sd-boot ist die Datei <code>/etc/os-release</code> notwendig, bei GRUB ist die Datei <code>/etc/default/grub.d/xxxx.cfg</code> ggf. zu erstellen oder zu ändern.
Linux OS auf Btrfs Dateisystem mit Unterstützung von snapper	-	o	sd-boot erstellt Menüeinträge nur einmalig bei der Installation der Kernel, gleichgültig aus welchem Subvolumen. Andere Subvolumen erhalten keinen Eintrag. GRUB ist je nach Distribution auf unterschiedliche, zusätzliche Software angewiesen.
siduction auf Btrfs Dateisystem mit Unterstützung von snapper	+	+	Das Paket <code>siduction-btrfs</code> erstellt Menüeinträge für sd-boot und GRUB nach einem Rollback. Der Standard Booteintrag bootet das Rollbackziel. Bei GRUB wird mit Hilfe des Pakets <code>grub-btrfs</code> das Untermenü <i>siduction snapshots</i> angezeigt.
Eine vollverschlüsselte HD	++	+	sd-boot reicht die Aufgaben an den Kernel und den User-Space weiter und ist dadurch effizienter. GRUB benötigt zusätzliche Software.

sd-boot spielt seine Vorteile bei einem einfachen Hardware Setup voll aus, aber scheitert bei Betriebssystemen auf mehreren Medien. Mit Unterstützung von `siduction-btrfs` eignet sich sd-boot auch gut für eine Installation in das Dateisystem Btrfs bei gleichzeitiger Verwendung von Snapper.

Grub wiederum ist universeller einsetzbar, dadurch schwergewichtig und benötigt trotzdem externe Software. Auch hier ist `siduction-btrfs` bei einer Installation in das Dateisystem Btrfs nützlich. Für einfache Hardware Setup ist GRUB überdimensioniert.

### 7.17.1 systemd-boot installieren

**Achtung**

Unbedingt eine Datensicherung auf einem externen Medium vornehmen.  
Es sind Arbeiten an Partitionen notwendig um sd-boot den Empfehlungen entsprechend zu installieren.

Die Anleitung basiert auf der Standardinstallation von *siduction* mit einer ESP (Efi System Partition), die unter `/boot/efi` eingehangen ist. Dies ist auch der Standard für Debian und viele Debian Derivate. Um zu sd-boot zu wechseln sind nur die zwei Pakete `systemd-boot` und `systemd-boot-efi` notwendig. Doch Vorsicht, zuerst sind einige Arbeiten am System erforderlich.

### 7.17.2 Vorbereitung des Systems

#### Partitionierung

Eine wesentliche Änderung von sd-boot gegenüber der Standardinstallation von siduction mit GRUB ist die Auslagerung des Verzeichnisses `/boot` in eine oder besser zwei separate Partitionen. Die Empfehlungen von sd-boot lauten:

- **Sowohl ESP als auch XBOOTLDR:**

*ESP*

Partitionstyp "EFI System", Part-GUID: C12A7328-F81F-11D2-BA4B-00A0C93EC93

Dateisystem: VFAT, eingehangen unter `/efi/`

Größe: 100 MB

*XBOOTLDR*

Partitionstyp "Linux extended boot", Part-GUID: BC13C2FF-59E6-4262-A352-B275FD6F7172

Dateisystem: Jedes, dass die UEFI-Implementierung lesen kann, eingehangen unter `/boot/`

Größe: mind. 1 GB

**sd.boot bug**

sd-boot kann seit systemd 258.2 Daten aus der XBOOTLDR Partition nur noch dann lesen, wenn die Partition mit vfat formatiert wurde.

[systemd-boot fails to use other file system than vfat](#)

- **Nur ESP:**

(Bedingt empfohlen, siehe unten.)

Partitionstyp "EFI System", Part-GUID: C12A7328-F81F-11D2-BA4B-00A0C93EC93

Dateisystem: VFAT, eingehangen unter /boot/

Größe: Mind. 1 GB

- **Nicht empfohlen:**

ESP eingehangen unter /boot/efi/

Hierzu übersetzte Zitate der [boot\\_loader\\_specification](#).

[Übersetzung Anfang]

*Hinweis: Diese Partitionen werden von allen Betriebssysteminstallationen auf derselben Festplatte gemeinsam genutzt, sodass alle denselben Ort für Einträge im Bootloader-Menü verwenden.*

[...]

*Das Einhängen des ESP in /boot/efi/, wie es traditionell gemacht wurde, wird nicht empfohlen. Eine solche verschachtelte Einrichtung erschwert eine Implementierung über direkte autofs-Einhängungen - wie sie beispielsweise von systemd implementiert werden -, da die Einrichtung des inneren autofs das äußere auslöst. Es wird empfohlen, die beiden Partitionen über autofs zu mounten. Da das einfache VFAT-Dateisystem eine geringe Datenintegrität aufweist, sollte es, wann immer möglich, nicht eingehängt werden.*

[...]

*Bei Systemen, bei denen die Firmware in der Lage ist, Dateisysteme direkt zu lesen, muss die ESP - und die GPT-XBOOTLDR-Partition sollte - ein Dateisystem sein, dass von der Firmware gelesen werden kann. Für die meisten Systeme bedeutet dies VFAT (16 oder 32 Bit). Anwendungen, die auf beide Partitionen zugreifen, sollten daher nicht davon ausgehen, dass ausgefeiltere Dateisystemfunktionen wie Symlinks, Hardlinks, Zugriffskontrolle oder Groß-*

*/Kleinschreibung unterstützt werden.*

[Übersetzung Ende]

### Mindestgröße

Hält man sich an die Empfehlungen von sd-boot und benutzt sowohl die ESP als auch die XBOOTLDR Partition, kann man die geringe Größe der ESP beibehalten und für die XBOOTLDR Partition ein leistungstärkeres Dateisystem verwenden. Notwendige Dateisystemtreiber für die XBOOTLDR Partition sind von [akeo.ie](http://akeo.ie) zu holen und nach `/efi/EFI/systemd/drivers/` zu kopieren.

Die XBOOTLDR Partition sollte mindestens 1 GB umfassen, da sd-boot alle Kernel und initrd darin doppelt ablegt. So wie es der Standard vorschreibt einmal direkt unter `/boot/` und ein weiteres Mal unter `/boot/<Kennung>/<Version>/`. Damit fallen für einen Kernel mit initrd 70 bis 100 MB an. Der Grund hierfür dürfte in der Verwendung von VFAT, das keine Symlinks unterstützt, liegen. Bei mehreren OS mit jeweils mehreren Kernel ist die Größe von 1 GB grenzwertig und die bisher üblichen 200 bis 300 MB der ESP sind völlig untauglich. Das zieht eine Änderung der Partitionierung nach sich.

Wir behalten die ESP in ihrer bisherigen Größe und erstellen an beliebiger Stelle auf dem gleichen Medium die XBOOTLDR Partition. Wie bereits erwähnt mindestens 1 GB groß, besser 2 GB. Wenn sich dabei die UUID einer vorhandenen Partition ändert, muss man die Datei `/etc/fstab` anpassen. Siehe: [Die fstab anpassen](#).

### Daten kopieren und Dateien anpassen

Wir öffnen ein Terminal und werden mit `su` zu ROOT.

Anschließend zeigt uns der `lsblk` Befehl für beide Partitionen den Typ Namen und die Typ GUID an.

```
# lsblk -n -o NAME,PARTTYPE,NAME,PARTTYPE /dev/nvme0n1p1 /dev/
/nvme0n1p4
nvme0n1p1 EFI System c12a7328-f81f-11d2-ba4b-00
a0c93ec93b
nvme0n1p4 Linux extended boot bc13c2ff-59e6-4262-a352-
b275fd6f7172
```

Sollte die Ausgabe nicht dem gezeigten Beispiel entsprechen, benutzen wir das Programm `cfdisk` um den Partitionstyp zu ändern.

Im nächsten Schritt legen wir neue Verzeichnisse an und kopieren das Verzeichnis `/boot`.

```
# cd /  
# mkdir /efi  
# mkdir /grub  
# cp -a /boot/* /grub/  
# umount /boot/efi/  
# rm -r /boot/*
```

Mit der Befehlsfolge sicherten wir auch die Daten der ESP.

Nun passen wir die Datei `/etc/fstab` an. (Zuvor eine Sicherungskopie erstellen.)

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

Von der Zeile mit dem Einhängpunkt `/boot/efi` erstellen wir eine Kopie und entfernen aus der ursprünglichen Zeile lediglich den String `"/boot"`.

In die Kopie schreiben wir die Daten für die XBOOTLDR Partition mit der Einhängung nach `/boot`.

Schließlich sollten die Änderungen so aussehen.

```
[...]  
UUID=FA62-156D          /efi    vfat    defaults 0 2  
UUID=<uuid_der_xbootldr> /boot  ext4    defaults 0 2  
[...]
```

Jetzt hängen wir die Partitionen ein und kopieren den Kernel mit Zubehör nach `/boot`. Dann erstellen wir noch das Verzeichnis für die Dateisystemtreiber der XBOOTLDR Partition.

```
# systemctl daemon-reload  
# mount /boot/  
# mount /efi/  
# cp -a /grub/vmlinuz-6.8.9-1-siduction-amd64 /boot/  
# cp -a /grub/initrd.img-6.8.9-1-siduction-amd64 /boot/  
# cp -a /grub/System.map-6.8.9-1-siduction-amd64 /boot/  
# cp -a /grub/config-6.8.9-1-siduction-amd64 /boot/  
# oder einfacher: cp -a /grub/*-6.8.9-1-siduction-amd64 /boot/  
# boot/
```

```
# mkdir -p /efi/EFI/systemd/drivers/
```

Zum Abschluss holen wir die Dateisystemtreiber von der Webseite [akeo.ie](http://akeo.ie) und speichern sie im erstellten Verzeichnis unterhalb `/efi/`. Auf Ausführrechte achten.

### Installation

Sind die Vorbereitungen abgeschlossen und wurde das Ergebnis z.B. mit den Befehlen `ls -l /boot`, `lsblk` oder `cfdisk` überprüft, reicht eine Kommandozeile aus um die zwei benötigten Pakete unserem System hinzuzufügen und sd-boot nach ESP und XBOOTLDR zu installieren.

```
# apt install systemd-boot-efi systemd-boot
[...]
Copied "/usr/lib/systemd/boot/efi/systemd-bootx64.efi" to "/efi/EFI/systemd/systemd-bootx64.efi".
Copied "/usr/lib/systemd/boot/efi/systemd-bootx64.efi" to "/efi/EFI/BOOT/BOOTX64.EFI".
Created "/boot/e5cc6ff820c1450c93a29d8723c78cd1".
! Mount point '/efi' which backs the random seed file is world accessible, which is a security hole!
! Random seed file '/efi/loader/random-seed' is world accessible, which is a security hole!
Random seed file /efi/loader/random-seed successfully installed (32 bytes).
Created EFI boot entry "Linux Boot Manager".
```

Im Firmware UEFI Menü finden wir jetzt an erster Stelle den *Linux Boot Manager*, der, wie die Ausgabe zeigt, in die ESP geschrieben wurde. Gleichzeitig hat sd-boot die Booteinträge für den *Linux Boot Manager* in der XBOOTLDR Partition erstellt.

Für jedes weitere OS auf dem Medium ist die 'Vorbereitung des Systems' und die 'Installation' notwendig.

#### 7.17.3 Konfiguration

Es gibt nur zwei Orte, an denen die Konfiguration für sd-boot liegt.

Für den *Linux Boot Manager* ist es die Datei `/efi/loader/loader.conf` und für

die *Menüeinträge* sind es die Dateien mit der Erweiterung `.conf` im Verzeichnis `/boot/loader/entries/`.

### Linux Boot Manager

Die Konfiguration des *Linux Boot Manager* betrifft nur die Punkte

- Anzeige des Menüs und Timeout - nein/ja/Dauer
- Konsole-Modus - Wert
- Editieren der Kernelzeile - ja/nein
- Standard Booteintrag festlegen - Wert

Bei nur einem System mit einem Kernel hat die Konfigurationsdatei folgenden Inhalt:

```
#timeout 3
#console-mode keep
```

Es wird ohne die Anzeige eines Menüs direkt in das Betriebssystem gebootet, denn 'timeout' ist auskommentiert. Der Grund: Bisher existiert nur ein einziger Menüpunkt. Nach entfernen des Kommentars erscheint das Menü einer siduction XFCE Installation.

```
siduction XFCE
Reboot Into Firmware Interface
```

Bei mehreren OS auf dem Medium erhält die Datei `/efi/loader/loader.conf` eine zusätzliche Zeile `default <Kennung>-*` um den Standardeintrag zu definieren.

```
timeout 3
#console-mode keep
default e5cc6ff820c1450c93a29d8723c78cd1-*
```

### Menüeinträge

Die Konfigurationsdateien liegen in dem Verzeichnis `/boot/loader/entries/` und ihr Name entspricht dem Format `<Kennung>-<Kernelversion>.conf`. sd-boot fügt dem Menü bei der Installation weiterer Kernel automatisch die Einträge hinzu. Mit einem zusätzlichen Kernel erhalten die Menüpunkte auch die Kernelversion.

```
siduction XFCE (6.8.10-1-siduction-amd64)
siduction XFCE (6.8.9-1-siduction-amd64)
Reboot Into Firmware Interface
```

Von **mehreren OS** auf unterschiedlichen Partitionen wird nur dasjenige automatisch im Menü aufgeführt, von dem aus sd-boot installiert wurde. Die dort vorhandenen Kernel erscheinen in der Auswahl. Installiert man in jedem OS sd-boot, finden diese Booteinträge auch ihren Weg in das Menü. Hier zum Beispiel UBUNTU.

```
siduction XFCE (6.8.10-1-siduction-amd64)
siduction XFCE (6.8.9-1-siduction-amd64)
Ubuntu 24.04 LTS
Reboot Into Firmware Interface
```

Sollte, aus welchen Gründen auch immer, ein Menüeintrag fehlen, kann die Generierung des Eintrags zu jeder Zeit erneut angestoßen werden. Es ist sicherzustellen, dass sich der Kernel, die vmlinuz und die config- Datei der entsprechenden Version im Verzeichnis `/boot/` befinden.

Aus dem entsprechenden OS heraus erstellt der Befehl

```
dpkg-reconfigure linux-image-6.8.9-1-siduction-amd64
```

auch den gewünschten Menüeintrag.

#### 7.17.4 GRUB entfernen

sd-boot kann zum jetzigen Zeitpunkt (08-2024) bei der Installation von siduction nicht als Standard Bootmanager ausgewählt werden. Sofern ausführliche Tests mit sd-boot erfolgreich verliefen, müssen wir GRUB aus unserem System entfernen um Fehler beim Update und Upgrade zu vermeiden.

##### Pakete

Wir öffnen ein Terminal, werden mit `su` zu ROOT und benutzen den Befehl `apt purge` um auch die Konfigurationsdateien zu entfernen.

```
# apt purge grub*
Paketlisten werden ...gelesen
```

```
Abhängigkeitsbaum wird ...aufgebaut
Statusinformationen werden ...eingelassen
Die folgenden Pakete werden ENTFERNT:
  grub-pc-bin* os-prober* grub-btrfs* grub2-common* grub-efi-
    amd64-bin*
  grub-common* grub-efi-ia32-bin* memtest86+* grub-pc*
0 aktualisiert, 0 neu installiert, 9 zu entfernen.
Nach dieser Operation werden 0 B Plattenplatz zusätzlich
benutzt.
Möchten Sie fortfahren? [J/n]
```

Nach dieser Aktion sind die von uns erstellten Sicherungskopien und zwei Verzeichnisse von GRUB übrig. Es handelt sich um das Verzeichnis `/grub/`, die Sicherungsdatei `/etc/fstab_*`, das Verzeichnis `/usr/share/grub/` und das Verzeichnis `/etc/default/grub.d/`.

Mit

```
# rm -r /grub/
# rm /etc/fstab_*
# rm -r /usr/share/grub/
# rm -r /etc/default/grub.d/
```

entledigen wir uns der Überbleibsel.

## UEFI bereinigen

In der Firmware existiert noch der siduction Booteintrag aus der Installation mit GRUB. Wir lassen uns im Terminal mit ROOT-Rechten den Inhalt anzeigen. (Gekürzt dargestellt.)

```
# efibootmgr
BootCurrent: 0001
Timeout: 0 seconds
BootOrder: 0001,0000,0002,2001,2002
Boot0000* siduction
Boot0001* Linux Boot Manager
Boot0002* EFI Hard Drive
```

```
Boot2001* EFI USB Device
Boot2002  EFI Network RC
```

Der erste Befehl entfernt den Eintrag von GRUB.

Der zweite Befehl löscht anschließend das zugehörige Verzeichnis aus [/efi/efi/](#).

```
# efibootmgr -b 0000 -B
# rm -r /efi/EFI/siduction/
```

### 7.17.5 systemd-boot und Btrfs

Das Dateisystem Btrfs bietet, besonders in Zusammenarbeit mit snapper, die Möglichkeit ein defektes System auf einen vorherigen Stand zurückzusetzen. In diesem Zusammenhang ist die von sd-boot verlangte Auslagerung des Verzeichnisses [/boot](#) in eine separate Partition hinderlich. Das Verzeichnis [/boot](#) ist ein wesentlicher Bestandteil des Betriebssystems und man sollte es im Zusammenhang mit Btrfs nicht in eine Partition oder ein Subvolumen auslagern. Denn dort wird es von Snapshots des Wurzeldateisystems nicht erfasst.

Trotzdem ist siduction mit dem Paket *siduction-btrfs* in der Lage nach einem Rollback für alle im Rollbackziel enthaltenen Kernel Menüeinträge zu erstellen. Außerdem bootet dann der Standard Booteintrag in das Rollbackziel. Menüeinträge für andere Snapshot stehen jedoch nicht zur Verfügung.

Das Paket *siduction-btrfs* ist nicht an einen bestimmten Bootmanager gebunden. Somit kann der Benutzer selbst entscheiden, ob er bei einer Installation des Systems auf Btrfs zu sd-boot wechseln möchte.

### 7.17.6 Weitere Informationen

[man systemd-boot](#)

[boot\\_loader\\_specification \(en\)](#)

[Dateisystem Treiber von akeo.ie](#)

Zuletzt bearbeitet: 2026-02-16

## 7.18 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.

### 7.18.1 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.

### 7.18.2 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.

### 7.18.3 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/journal@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.

### 7.18.4 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 Prioritätsbereichen. 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 Journal-einträgen 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.

- `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

ne 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.

### 7.18.5 journalctl beherrschen

Wie oben unter Rechte beschrieben, kannst Du das Journal als einfacher User benutzen. Hier sind einige Beispiele:

Befehl	Anzeige
<code>journalctl</code>	das volle Journal aller User, älteste Einträge zuerst
<code>journalctl -r</code>	wie zuvor, neueste Einträge zuerst
<code>journalctl -b</code>	das Protokoll des letzten Bootvorgangs
<code>journalctl -b -1 -k</code>	vom vorletzten Bootvorgang (-1) alle Kernelmeldungen
<code>journalctl -b -p err</code>	limitiert auf den letzten Boot und die Priorität ERROR
<code>journalctl --since=yesterday</code>	das Journal seit gestern
<code>journalctl /dev/sda</code>	das Journal der Gerätedatei /dev/sda
<code>journalctl /usr/bin/dbus-daemon</code>	alle Logs des D-Bus-Daemon
<code>journalctl -f</code>	Liveansicht des Journal (früher: <code>tail -f /var/log/messages</code> )

Die Option “--list-boots\*” gibt die entsprechende Liste aus.

```
# journalctl --list-boots --no-pager
[...]
-50 8fc07f387... Sun 2021-02-28 11:07:05 CET-Sun [...] CET
-49 aa49cb3af... Mon 2021-03-01 17:49:58 CET-Mon [...] CET
-48 3a6e55a4a... Tue 2021-03-02 12:18:46 CET-Tue [...] CET
-47 a46150a19... Wed 2021-03-03 11:06:29 CET-Wed [...] CET
-46 d42ed8b05... Thu 2021-03-04 10:59:56 CET-Thu [...] CET
-45 566f65991... Thu 2021-03-04 19:53:52 CET-Thu [...] CET
-44 8e2da4a61... Fri 2021-03-05 10:15:18 CET-Fri [...] 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=
_CAP_EFFECTIVE=
_CMDLINE=
_CODE_FILE=
_CODE_FUNC=
_CODE_LINE=
_COMM=
_COMMAND=
_CONFIG_FILE=
_CONFIG_LINE=
_CURRENT_USE=
_CURRENT_USE_PRETTY=
_DISK_AVAILABLE=
_DISK_AVAILABLE_PRETTY=
_DISK_KEEP_FREE=
_DISK_KEEP_FREE_PRETTY=
_ERRNO=
_EXE=
_EXECUTABLE=
_EXIT_CODE=
_EXIT_STATUS=
_FSUID=
_GID=
_GLIB_DOMAIN=
_GLIB_OLD_LOG_API=
_HOSTNAME=
_INVOCATION_ID=
_JOB_ID=
_JOB_RESULT=
_JOB_TYPE=
_JOURNAL_NAME=
_JOURNAL_PATH=
_KERNEL_DEVICE=
_SELINUX_CONTEXT=
_SESSION_ID=
_SHUTDOWN=
_SLEEP=
_SOURCE_MONOTONIC_TIMESTAMP=
_SOURCE_REALTIME_TIMESTAMP=
_STREAM_ID=
_SYSLOG_FACILITY=
_SYSLOG_IDENTIFIER=
_SYSLOG_PID=
_SYSLOG_RAW=
_SYSLOG_TIMESTAMP=
_SYSTEMD_CGROUP=
_SYSTEMD_INVOCATION_ID=
_SYSTEMD_OWNER_UID=
_SYSTEMD_SESSION=
_SYSTEMD_SLICE=
_SYSTEMD_UNIT=
_SYSTEMD_USER_SLICE=
_SYSTEMD_USER_UNIT=
_THREAD_ID=
_TIMESTAMP_BOOTTIME=
_TIMESTAMP_MONOTONIC=
_TRANSPORT=
_UDEV_DEVNODE=
_UDEV_SYSNAME=
_UID=
_UNIT=
_UNIT_RESULT=
_USER_ID=
_USER_INVOCATION_ID=
_USERSPACE_USEC=
_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-udev
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 --
Mar 10 16:27:22 pc1 su[105197]: (to root) siduser on pts/1
Mar 10 16:27:22 pc1 su[105197]: pam_unix(su:session):
    session opened for user root(uid=0) by (uid=1000)
Mar 10 17:54:33 pc1 su[105197]: pam_unix(su:session):
    session closed for user root
```

```
-- boot 37b19f6321814620be1ed4deb3be467f --
Mar 10 17:56:35 pc1 su[3381]: (to root) siduser on pts/1
Mar 10 17:56:35 pc1 su[3381]: pam_unix(su:session):
    session opened for user root(uid=0) by (uid=1000)
Mar 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.

### 7.18.6 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.

Zuletzt bearbeitet: 2021-11-29