



HOCHSCHULE MITTWEIDA  
UNIVERSITY OF APPLIED SCIENCE

LEHRBRIEF

für das Modul

# **Betriebssysteme**

Autor: Prof. Ronny Bodach

Bearbeitungsstand: 22.06.2022

# Hinweise

Herausgeber:

©2022 Hochschule Mittweida

Hochschule Mittweida - University of Applied Sciences

Fakultät Computer- und Biowissenschaften

Technikumplatz 17

09648 Mittweida

1. Auflage (23.03.2022)

Redaktionelle Bearbeitung: Prof. Ronny Bodach

Das Werk einschließlich seiner Teile ist urheberrechtlich geschützt. Jede Verwendung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung der Verfasser unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

# Inhaltsverzeichnis

1	Betriebssystem Linux als Client .....	4
1.1	Linux Historie, Kernel, Distributionen .....	4
1.1.1	Open Source Lizenzen .....	4
1.1.2	Linux-Kernel.....	6
1.1.3	Distributionen.....	9
1.1.4	Zusammenfassung.....	15
1.2	Linux Grundlagen.....	16
1.2.1	Installation.....	16
1.2.2	Bootvorgang .....	22
1.2.3	FHS - Verzeichnisstruktur unter Linux .....	23
1.2.4	Verlinkung von Dateien .....	25
1.2.5	Zusammenfassung.....	26
1.3	Linux Grundlagen II.....	27
1.3.1	Netzwerk-Konfiguration .....	27
1.3.2	Dateisysteme Überblick.....	28
1.3.3	Benutzer und Gruppen .....	33
1.3.4	Nutzungsverlauf .....	37
1.3.5	Zusammenfassung.....	41
1.4	Linux Anwendungen im Terminal I.....	42
1.4.1	Linux Shell.....	42
1.4.2	Navigation.....	42
1.4.3	Dateiinhalt .....	48
1.4.4	Rechteverwaltung .....	53
1.4.5	Zusammenfassung.....	59
1.5	Linux Anwendungen im Terminal 2.....	60
1.5.1	Systemüberblick .....	60
1.5.2	Konsoleneditor .....	64
1.5.3	Datenträger .....	67
1.5.4	Netzwerk .....	71
1.5.5	Skripte.....	74
1.5.6	Zusammenfassung.....	83
1.6	Linux Paket- und Dienstmanagement .....	84
1.6.1	Softwareinstallationswege .....	84
1.6.2	Paketverwaltung.....	86
1.6.3	Binaries .....	93

1.6.4	Source Code.....	94
1.6.5	Dienstverwaltung .....	95
1.6.6	Zusammenfassung.....	96
1.7	Linux Selbsthilfe.....	97
1.7.1	Manual Pages .....	97
1.7.2	Herstellerwebseite .....	101
1.7.3	Foren und Wikis.....	105
1.7.4	Blogs und Webseiten.....	111
1.7.5	Zusammenfassung.....	114
1.8	Namespaces und Git.....	115
1.8.1	Git .....	115
1.8.2	Namespaces.....	139
1.8.3	Docker.....	140
1.9	Linux Sicherheit .....	142
1.9.1	Grundprinzipien der IT-Sicherheit .....	142
1.9.2	Programmüberwachung.....	143
1.9.3	Netzwerksicherheit.....	144
1.9.4	Rechteverwaltung .....	152
1.9.5	Fernzugriff .....	153
1.9.6	Dateisicherheit .....	156
1.9.7	Zusammenfassung.....	158



# 1 Betriebssystem Linux als Client

## 1.1 Linux Historie, Kernel, Distributionen

Unix wurde bis Version 7, welche 1979 erschien, von AT&T entwickelt. Es ist kostenlos und ohne rechtliche Einschränkungen verfügbar, sodass es vor allem im Universitätsumfeld verteilt ist. Unix wurde zum Selbstkostenpreis vertrieben.

1983 bot das GNU-Projekt einen Gegenpol zur Kommerzialisierung. Die GPL-Lizenz gewährleistete hier die zukünftige Freiheit im Sinne der freien Zugänglichkeit. Das GNU-Projekt besteht aus kleinen alltäglichen Programmen.

Damals wurde Berkeley Software Distribution (BSD) als Kernel eingesetzt.

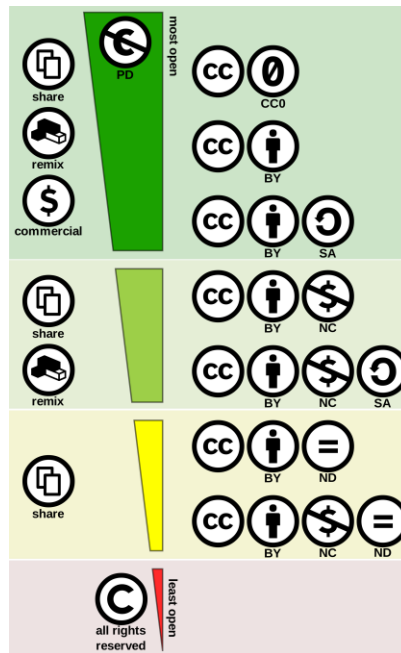
### 1.1.1 Open Source Lizenzen

Es gibt mehrere Hauptgruppen von Open Source Lizenzen. Dazu zählen

- Creative Common License, bei welcher es verschiedene Abstufungen gibt
- GNU Public License mit drei verschiedenen Versionen
- BSD License mit drei Versionen
- Apache License mit zwei Versionen
- MIT License
- Eclipse Public License
- Do-What-The-Fuck-You-Want License
- The Unlicense

#### 1.1.1.1 Creative Common License

Die Creative Common lässt sich in logische Gruppen aufteilen. Dazu gehören die Gruppen keine Rechte verbleiben (CC-0), die Verteilung unter der Namensnennung ist erlaubt (BY), die Weitergabe unter gleichen Bedingungen (Lizenerhaltung) und Änderungen sind erlaubt (SA), die kommerzielle Nutzung ist nicht erlaubt (NC), die Bearbeitung ist nicht erlaubt (ND) und es werden keine Rechte gewährt (C).



Als Beispiel CC BY-NC-SA bedeutet, dass die Verteilung unter der Namensnennung erlaubt ist, die kommerzielle Nutzung ist verboten und Änderungen unter gleicher Lizenz sind erlaubt.

#### 1.1.1.2 GNU Public License

Bei GPLv1 ist der Code sowohl kopierbar als auch veränderbar. Der Code muss auch wieder veröffentlicht werden. Bei dem Code handelt es sich um menschenlesbaren Code, welcher als GPLv1 vertrieben werden muss (auch die Libraries). Libraries mit einer restriktiven Lizenz sind nicht verwendbar.

GPLv2 wird beim Linux Kernel verwendet. Wie auch schon bei GPLv1 ist der Code kopierbar, veränderbar, muss veröffentlicht werden, ist menschenlesbar und als GPLv2 vertrieben werden. In diesem Fall sind jedoch die Libraries mit restriktiver Lizenz verwendbar.

GPLv2.1 ist die GNU Lesser General Public License (LGPL) und ist fast identisch zur Version 2 (GPLv2). Jedoch erlaubt GPLv2.1 das Upgrade auf die Versionen 3 und 3.1.

GPLv3 basiert auf der Version 2. Sie ist kompatibel zu Software-patents und anderen freien Software-Lizenzen.

GPLv3.1 basiert auf der Version 2.1. Es beinhaltet die gleichen Anpassungen wie die Version 3.

## 1.1.2 Linux-Kernel

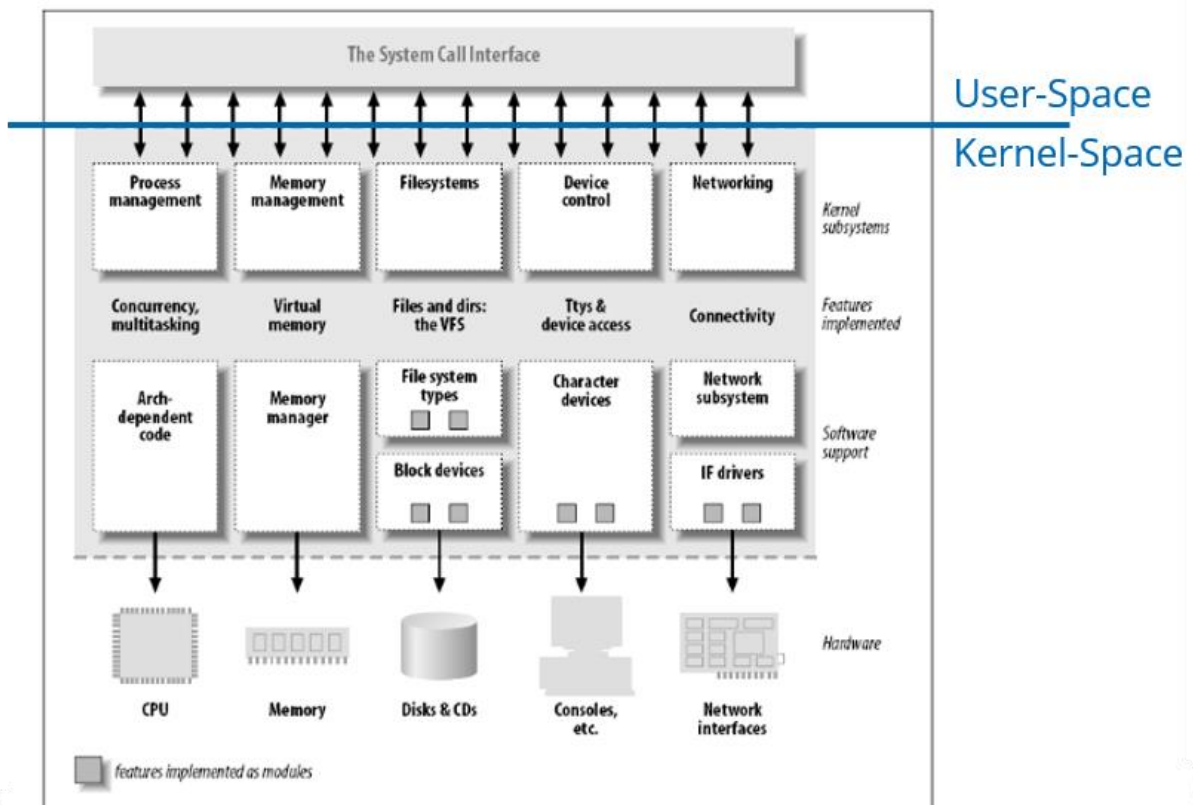
Der Linux Kernel wurde ursprünglich von Linus Torvalds entwickelt, 1991 in einem Minix-Forum vorgestellt und im selben Jahr über FTP-Server verteilt. Er gilt unter der GNU Public License Version 2 (GPLv2) als quelloffen und ist in C programmiert. Die Versionsverwaltung wird mit BitKeeper durchgeführt, wobei ab 2005 Git für die Versionsverwaltung verwendet wurde. Die aktuelle Version ist 5.17.2, welche am 08.04.2022 veröffentlicht wurde. Eine Dokumentation des Kernels findet man unter <https://linux-kernel-labs.github.io/refs/heads/master/lectures/intro.html>.

### 1.1.2.1 Linux Kernel Aufgaben

Der Linux Kernel ist für verschiedene Aufgaben zuständig. Er bildet die Schnittstelle zur Hardware und abstrahiert den Zugriff. Weiterhin verwaltet er Ressourcen wie die CPU-Zeit, den Arbeits- und Festplattenspeicher sowie den Netzwerkzugriff. In Bezug auf die Rechte verwaltet der Benutzer-, Gruppen und Zugriffsrechte. Der Linux Kernel unterstützt folgende Architekturen:

- x86
- Intel 64, AMD 64
- ARMv4 und höher
- OpenRISC
- PowerPC
- Sun/Oracle SPARC
- MIPS
- Motorola 68020
- Atmel AVR32
- Hitachi H8/300
- ... alle anderen

### 1.1.2.2 Kernel Architektur



#### System Call Interface

Das System Call Interface bildet die Schnittstelle für Anwendungen und realisiert den Übergang zwischen User- und Kernel-Mode. Die Kernaufgaben des System Call Interfaces sind das Sammeln von Setup-Informationen, die Identifizierung von System Calls und Parametern, das Auslösen des Kernel-Mode-Switches, die Weitergabe an ein zuständiges Modul und das Entgegennehmen von Ergebnissen. Beispiel: `fork()` -> Process Management

#### Process Management

Das Process Management verwaltet Prozesse und Threads. Weiterhin ist es für das Context Switching, sprich den Wechsel zwischen User- und Kernel-Mode, das Blockieren und Freigeben (Warten auf Ressourcenantwort), die Verwaltung des Prozess-Kontextes und die Namespaces (Container wie Docker, LXC, ...) zuständig.

## Memory Management

Das Memory Management realisiert die Verwaltung des Arbeitsspeichers in Bezug auf die Reservierungen (`malloc()`) und die Freigaben (`free()`). Zusätzlich ist das Memory Management für die Zuordnung des virtuellen Speichers zum physischen Speicher, die page allocation (Seitenzuteilung), die small allocation (Page in Teile aufteilen), die Page-Fault Bearbeitung und für die Zugriffsrechte auf den Speicher verantwortlich.

## File System

Das File System stellt das Linux Virtual File System bereit. Weiterhin bindet es Speicherressourcen ein und ordnet physische Datenträger zu Dateien zu. Das File System kann Dateien schreiben, lesen und dessen Attribute (Rechte) auswerten und setzen.

## Device Control

Device Control ist für den direkten Zugriff auf die Hardware verantwortlich. Außerdem liest es Debug-Informationen wie den Prozessorstatus (Register), den Memory-Content, die Software-Interrupts und den Performance Counter aus.

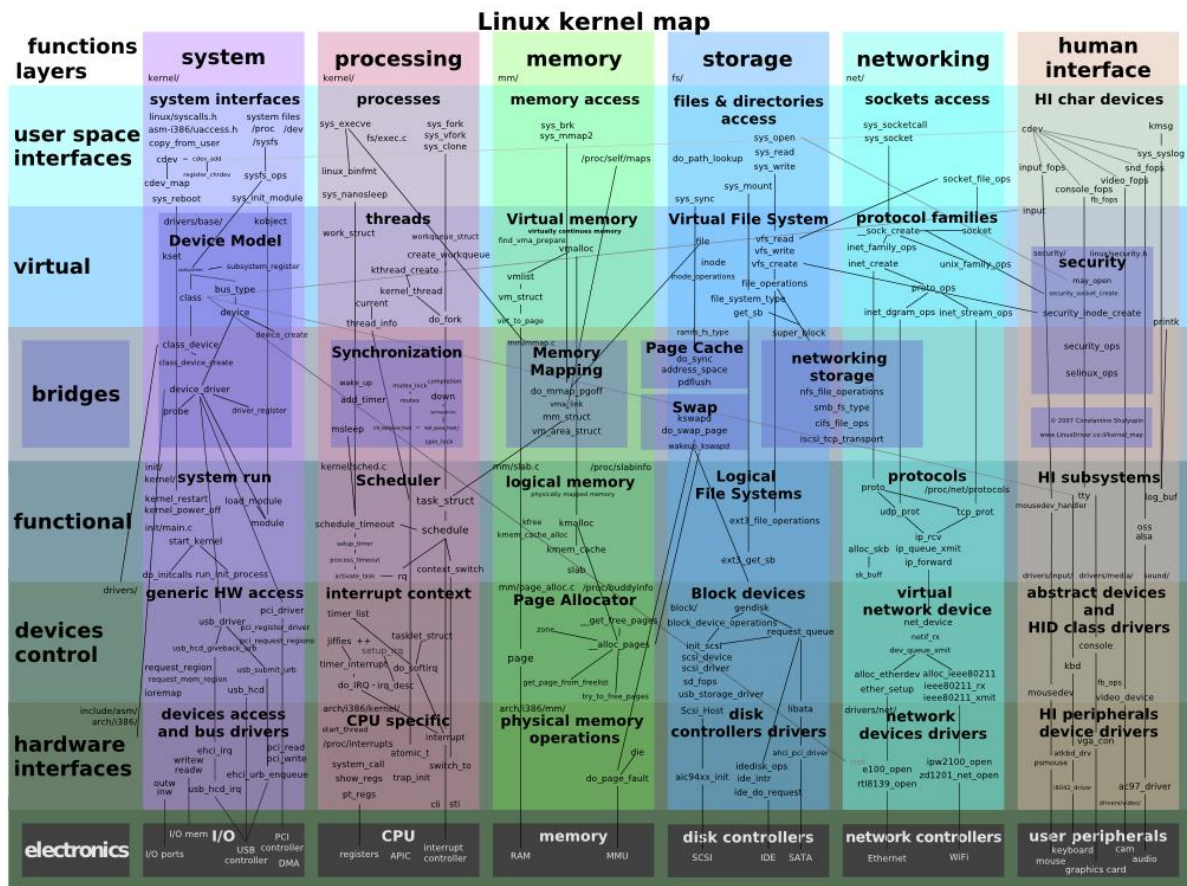
## Networking

Networking bildet die Netzwerkschnittstelle für den User. Es führt das TCP/IP-Stack Handling durch, indem es TCP-Verbindungen aufbaut, UDP-Datagramme versendet und empfängt und Sockets einrichtet. Außerdem ist Networking für die Netzwerkfirewall (`netfilter`) und die systeminterne Netzwerkkommunikation zuständig.

## Treiber

Die Treiber abstrahieren die Hardware. Weiterhin realisieren sie Zugriffe wie das Schreiben, das Lesen und hardware-spezifische Operationen auf die Hardware.

### 1.1.2.3 Linux Kernel im Detail



Die Abbildung findet man unter

[http://3.bp.blogspot.com/\\_wdYQghStev8/TUUwX8vHLyI/AAAAAAAAABA/LgnEx\\_yttc8/s1600/Linux\\_kernel\\_map.png](http://3.bp.blogspot.com/_wdYQghStev8/TUUwX8vHLyI/AAAAAAAAABA/LgnEx_yttc8/s1600/Linux_kernel_map.png)

### 1.1.3 Distributionen

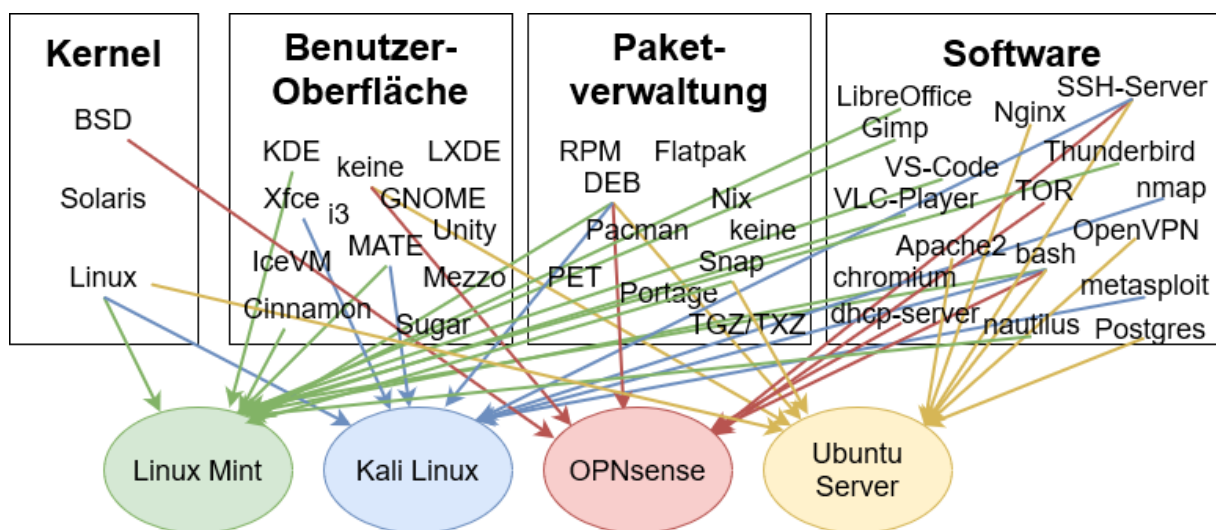
Eine Distribution ist eine Softwarezusammenstellung oder eine Softwareverteilung. In Bezug auf Linux handelt es sich hierbei um ein fertig nutzbares Betriebssystem, auf welchem bereits einige Softwarekomponenten vorinstalliert und einige im Repository (Online-Bibliothek/Shop) vorhanden sind. Die Distribution beinhaltet den Kernel mit Treibern, die Software, die Standardkonfigurationen und die Benutzeroberflächen.

### 1.1.3.1 Distributionen Spezialisierungen

Bei den Distributionen gibt es verschiedenste Spezialisierungen. So kann zwischen Server und Desktop, zwischen neuer und alter Hardware, zwischen Stabilität und neuester Software, zwischen geringem und großem Umfang oder auch zwischen Spielen, Business, Bildung, Multimedia und Wissenschaft unterschieden werden.

Weitere Spezialisierungen gibt es in Bezug auf das Netzwerk (Firewall, Router, ...), die Sprachgruppe, die Hardwarearchitektur (x64, ARM, PowerPC, riscv64, ...), den Langzeitsupport (Sicherheitsupdates) oder auch in Bezug auf die Forensik und Pentesting.

### 1.1.3.2 Distributionserstellung



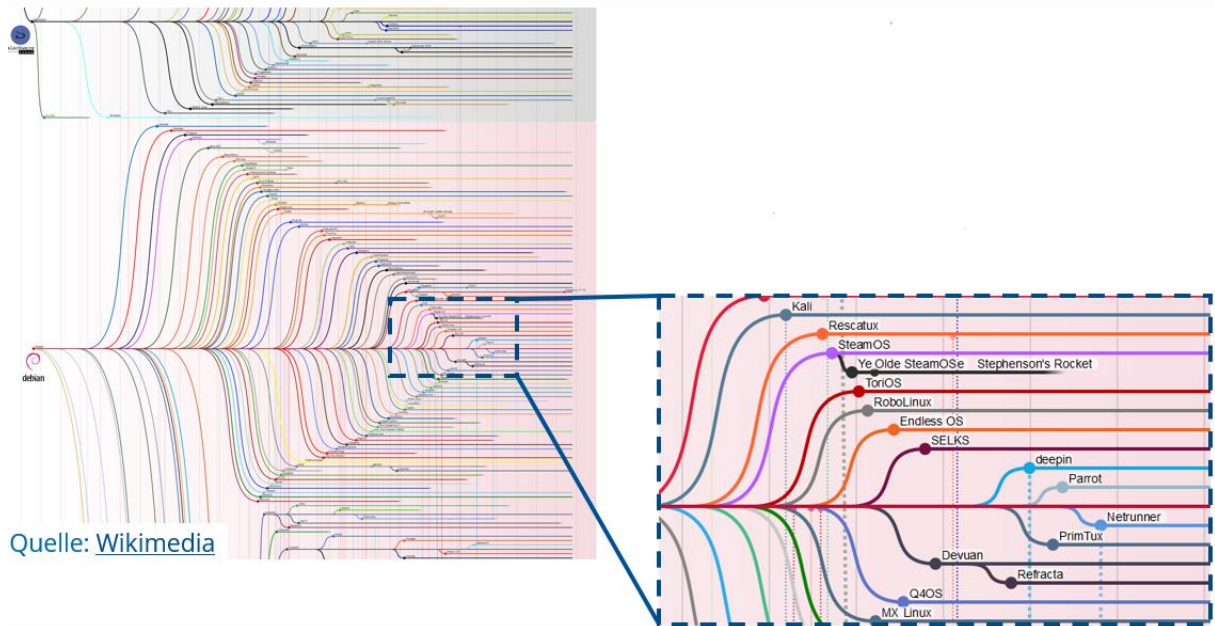
### 1.1.3.3 Distrowatch.com

Distrowatch.com ist ein Webforum für Linux Distributionen. Es bietet eine Auflistung der Distributionen sowie eine Kurzbeschreibung von diesen, Benutzererfahrungen, Versionen von Softwarekomponenten, Spezialisierungen, ein Beliebtheitsranking, Verlinkungen zu Distro-Webseiten und Ankündigungen von neuen Distributionen.

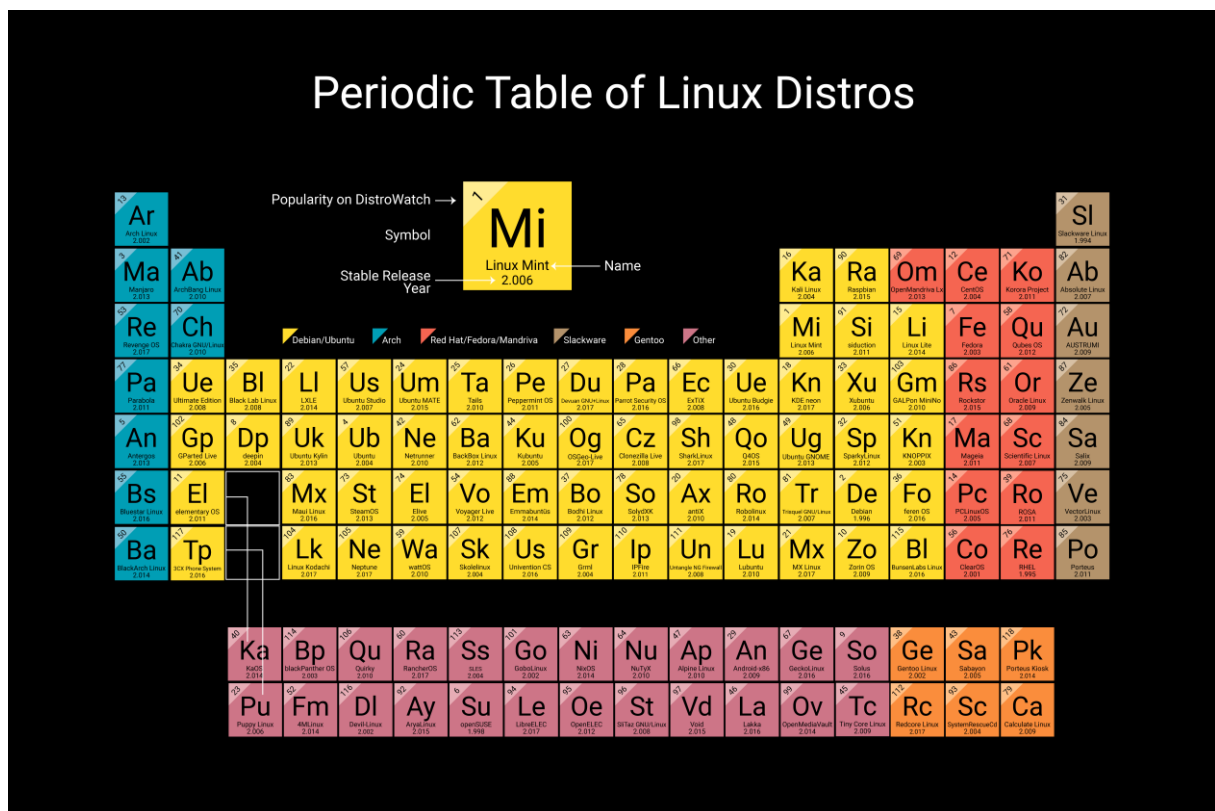
Laut aktuellem Stand vom 11.04.2022 werden 931 Distributionen gelistet, wovon 269 aktiv sind. 51 sind vernachlässigt, da für sie schon zwei Jahre lang keine neue Version herausgekommen ist. 611 Distributionen sind veraltet bzw. inaktiv, weitere 178 auf der Warteliste und 32 warten auf eine Beurteilung.



### 1.1.3.4 Stammbaum der Linux Distributionen



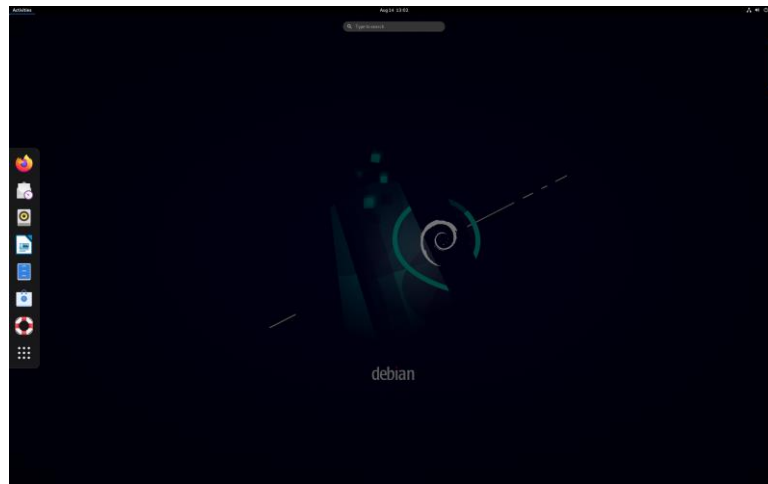
### 1.1.3.5 Periodic Table of Linux Distros





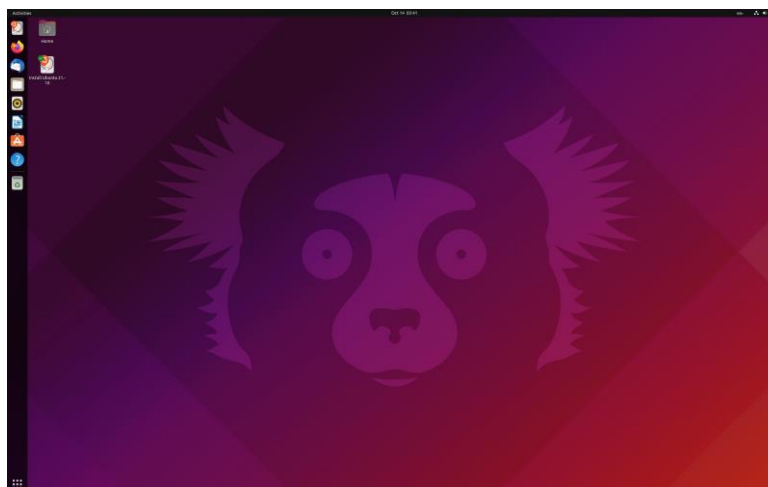
### 1.1.3.6 Debian

Debian wird weltweit verwendet und ist kostenlos. Als Desktop Oberfläche wird beispielsweise GNOME verwendet und das Release Model ist fest. Unterstützte Prozessor Architekturen sind x86, x64, ARM64, PowerPC, MIPS und s390x. Integrierte Office Suiten sind Calligra, GOffice und Libreoffice. Die Philosophie von Debian ist, dass die Stabilität oberste Priorität hat und ältere Software zur Gewährleistung der Stabilität eingesetzt wird.



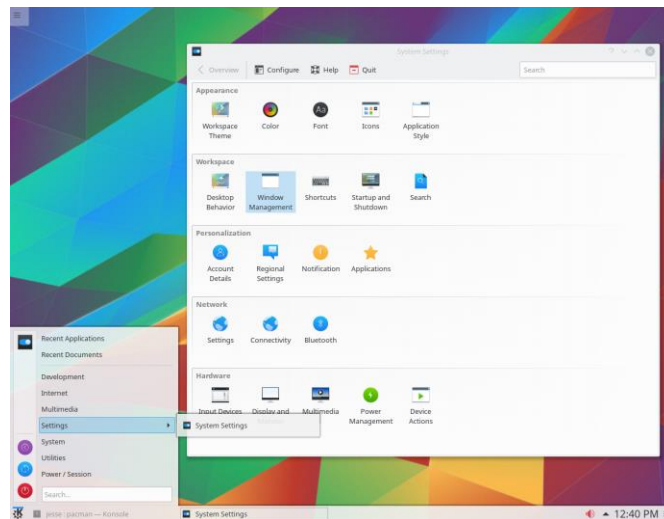
### 1.1.3.7 Ubuntu

Ubuntu kommt ursprünglich von der Isle of Man (UK) und ist ebenfalls kostenlos. Auch hier ist das Release Model fest und als Desktop Oberfläche werden Unity und GNOME verwendet. Unterstützte Prozessor Architekturen sind x64, armhf, PowerPC, Riscv und s390x. Als Office Suite ist LibreOffice integriert. Die Philosophie von Ubuntu ist die Nutzerorientierung, die Sprachunterstützung, es ist für Linux-Anfänger geeignet und vor allem Linux-Mint setzt seinen Fokus auf Windows Nutzer.



### 1.1.3.8 Arch

Arch kommt aus Kanada und ist ebenfalls kostenlos. Das Release Model ist rollend und Arch unterstützt einzig x86. Als Desktop Oberfläche werden beispielsweise Cinnamon oder Enlightenment verwendet. Integrierte Office Suiten sind GOffice, Calligra und LibreOffice. Die Philosophie von Arch ist, dass es eher für fortgeschrittene Nutzer geeignet ist und immer neueste Software-Versionen verwendet.



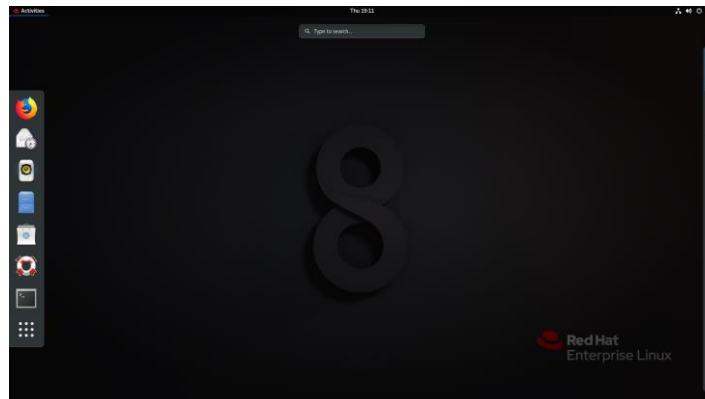
### 1.1.3.9 Fedora

Fedora kommt aus den USA und ist kostenlos verfügbar. Es unterstützt die Architekturen x64 und ARM64. Das Release Model ist fest und als Desktop Oberfläche werden unter anderem GNOME und i3 genutzt. Die integrierte Office Suite ist LibreOffice. Fedoras Philosophie ist, dass Open Source Software eine Führungsrolle einnimmt. Außerdem setzt es den Fokus auf Innovation und spezialisierte Versionen für Gaming, Security, Design, Forschung oder Ähnliches.



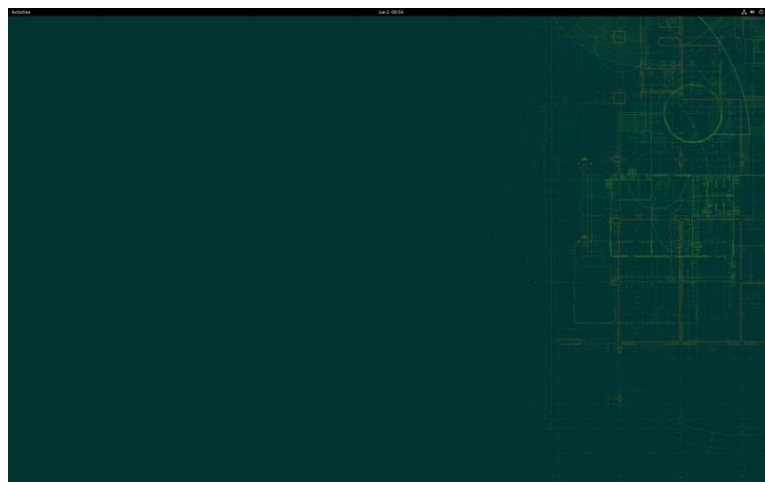
### 1.1.3.10 Red Hat

Red Hat kommt aus den USA und kostet zwischen 99 und 500€. Das Release Model ist fest und als Desktop Oberfläche wird GNOME verwendet. Unterstützte Prozessor Architekturen sind x86, x64, ARM64, PowerPC, IBM Z und s390. Als Office Suite wird LibreOffice verwendet. Red Hat setzt auf die kommerzielle Nutzung sowie dass es Open Source und einfach zu administrieren ist. Weiterhin bietet es technischen Support.



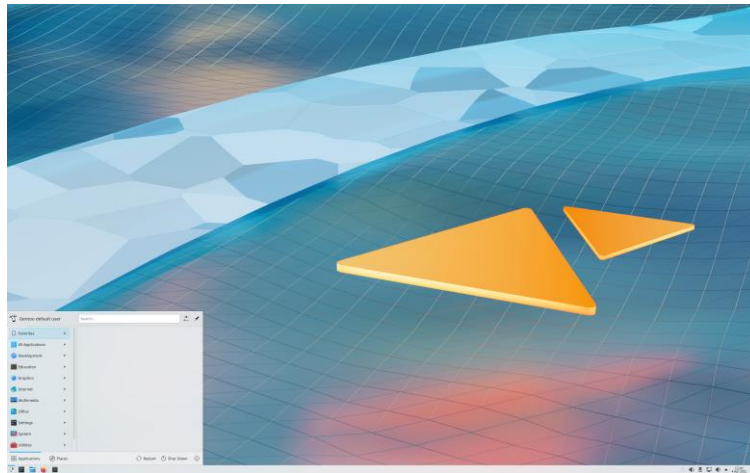
### 1.1.3.11 openSUSE

openSUSE stammt aus Deutschland und ist kostenlos. Das Release Model ist entweder rollend oder fest. Die Desktop Oberfläche ist beispielsweise Cinnamon und integrierte Office Suites sind Calligra, GOffice und LibreOffice. openSUSE unterstützt x86, x64, ARM64 und PowerPC. Die Philosophie von openSUSE ist, dass es einfach zu nutzen ist, es die Softwarebereitstellung für Entwickler vereinfacht und eigene Konfigurationstools wie yum mitbringt.



### 1.1.3.12Gentoo

Gentoo stammt aus den USA und ist kostenlos. Das Release Model ist rollend und als Desktop Oberfläche wird beispielsweise KDE Plasma verwendet. Unterstützte Prozessorarchitekturen sind X64, ARM, PowerPC und viele alte CPUs (i486). Integrierte Office Suites sind GOffice, Calligra und LibreOffice. Die Philosophie von Gentoo ist, dass es sich auf Entwickler und Netzwerkspezialisten spezialisiert und sich an der BSD-Verwaltung orientiert. Außerdem setzt Gentoo auf eine eigene Paketverwaltung (portage).



### 1.1.4 Zusammenfassung

Sie haben die Vorgeschichte von Linux kennengelernt und haben nun einen groben Überblick über Open-Source Lizenzen.

Über den Linux Kernel wissen Sie nun den grundlegenden Aufbau. Ihnen ist bekannt, aus welchen Grundbausteinen der Linux-Kernel besteht und welche Aufgaben diese erfüllen.

Der Begriff Distribution im Zusammenhang von Linux sollte ihnen bekannt sein. Es sollte Ihnen klar geworden sein, dass es nicht das eine Linux gibt. Sie kennen nun einige Eckpunkte, was eine Distribution ausmacht und nach welchen Kriterien man eine passende Distribution aussucht.

Die Hauptgruppen der Distributionen wurden Ihnen heute vorgestellt. Dazu gehören Debian, Ubuntu, Arch, Red Hat, Fedora, openSUSE und Gentoo.

## 1.2 Linux Grundlagen

### 1.2.1 Installation

Vor der Installation muss eine Distribution gewählt werden. Zur Auswahl stehen Debian, Ubuntu, Linux Mint, Arch, Manjaro oder Fedora. Weiterhin wählt man die x86 (32 Bit), die x64 (64 Bit) oder die ARM64 (64 Bit) Architektur. In Bezug auf die Oberfläche stehen Cinnamon, KDE, Xfce, MATE, GNOME oder Unity zur Auswahl. Zusätzlich wählt man die Installationsweise und die Downloadart (Direkt oder Torrent).

#### 1.2.1.1 Installationsmöglichkeiten

Für die Installation von Linux gibt es drei verschiedene Möglichkeiten. Der Live-Modus wird über einen USB-Stick, eine DVD oder eine CD gestartet. Hierbei werden keine Veränderungen auf der Festplatte durchgeführt. Weiterhin sind die Änderungen nicht persistent, sprich alle Änderungen sind nach dem Ausschalten gelöscht.

Eine weitere Möglichkeit ist die Installation. Die Installation kann aus dem Live-Modus oder aus dem Boot-Menü heraus durchgeführt werden. Hierbei werden alle Daten auf dem Datenträger abgelegt. Dies nimmt viel Speicherplatz ein, aber die Daten sind auch offline verfügbar.

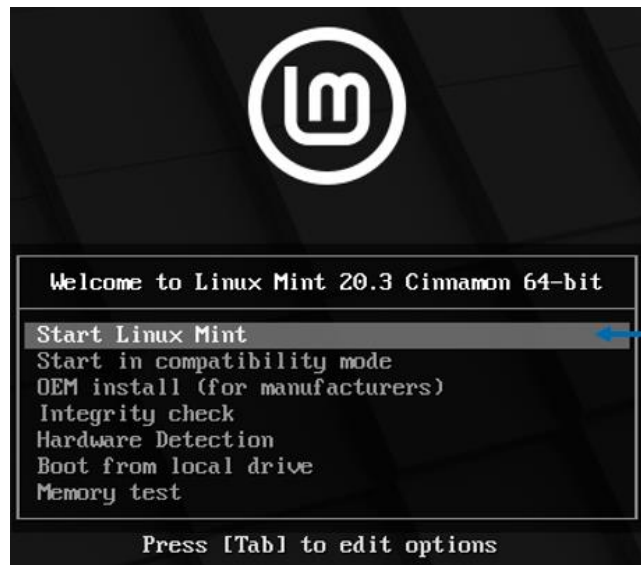
Die letzte Möglichkeit ist Netinstall. Hierbei wird ein Mini-OS auf das Startmedium aufgespielt. Die Software wird während der Installation aus dem Internet geladen.

#### 1.2.1.2 Linux am Beispiel „Linux Mint“

Für die beispielhafte Installation wird Linux Mint 20.3 mit Cinnamon als GUI verwendet. Der Download des ISO erfolgt direkt über Netcologne GmbH (<https://mirror.netcologne.de/linuxmint/iso/stable/20.3/linuxmint-20.3-cinnamon-64bit.iso>). Nach dem Download sollte die SHA256-Prüfsumme abgeglichen werden. Die Linux Mint Webseite findet man unter <https://linuxmint.com/edition.php?id=292>.

Linux Mint ist sehr ähnlich zu Ubuntu. Die Benutzeroberfläche orientiert sich an Windows, womit es vor allem für Linux Einsteiger geeignet ist. Weiterhin bietet Linux Mint einen sehr guten Hardware-Support in Bezug auf Drucker, WLAN oder Ähnliches.

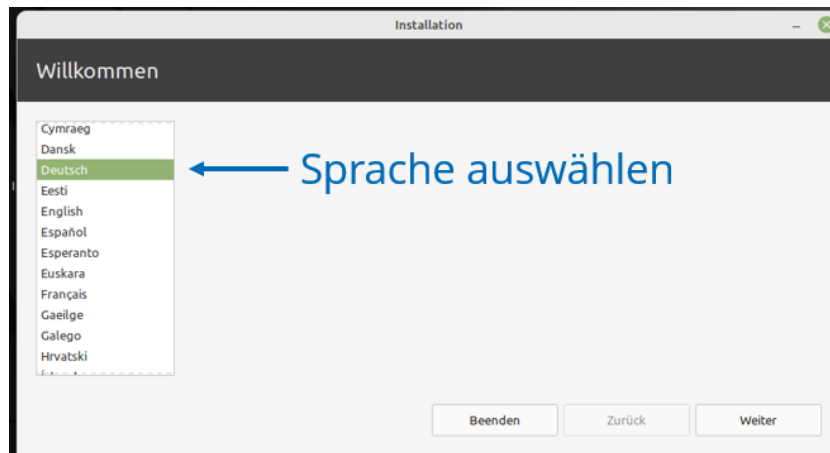
Beginnend startet man den Live Modus.



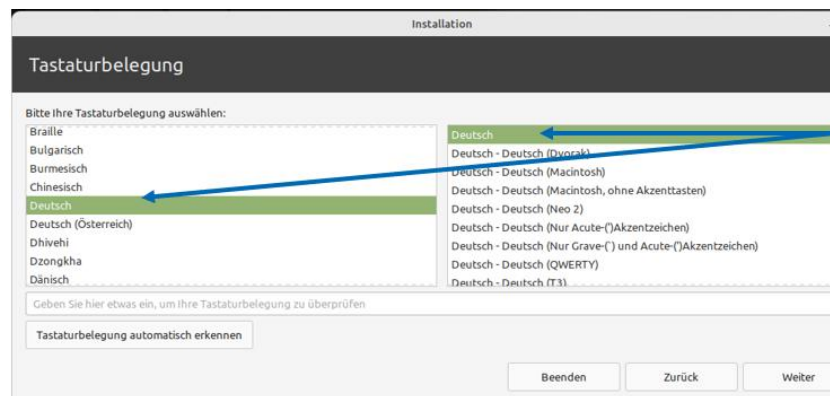
Im Anschluss kann man die Installation starten.



Zu Beginn der Installation wird die Sprache ausgewählt.



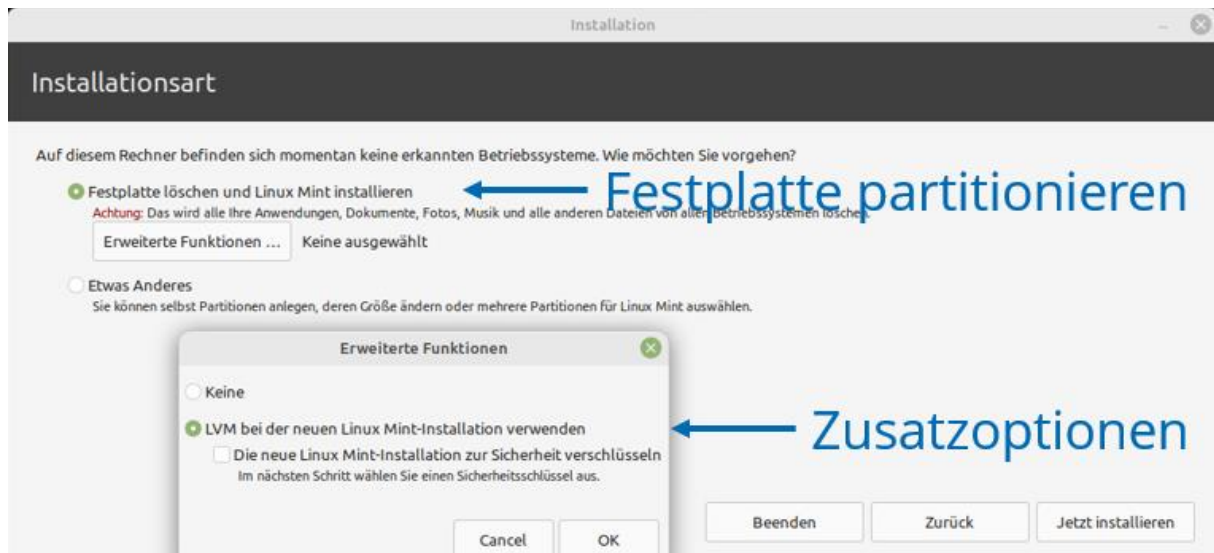
Anschließend wählt man das Tastaturlayout.



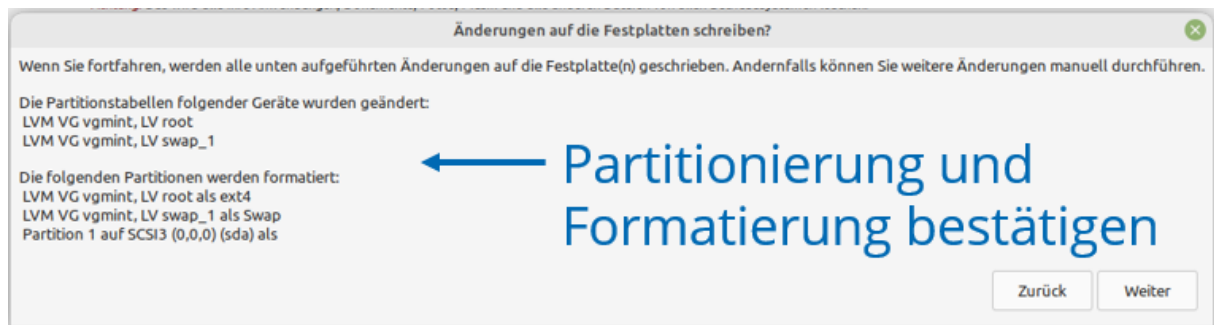
In Bezug auf die Multimedia-Codecs wählt man aus, dass proprietäre Media-Codecs mit installiert werden. Alternativ kann man dies auch später mit dem Befehl `sudo apt install mint-meta-codecs` durchführen.



Für die Installationsart wählt man den Punkt „Festplatte löschen und Linux Mint installieren“, um die Festplatte zu partitionieren. Über den Button „Erweiterte Funktionen“ öffnet sich ein Fenster mit Zusatzoptionen. Hier wählt man den Punkt „LVM bei der neuen Linux Mint-Installation zur Sicherheit verschlüsseln“.

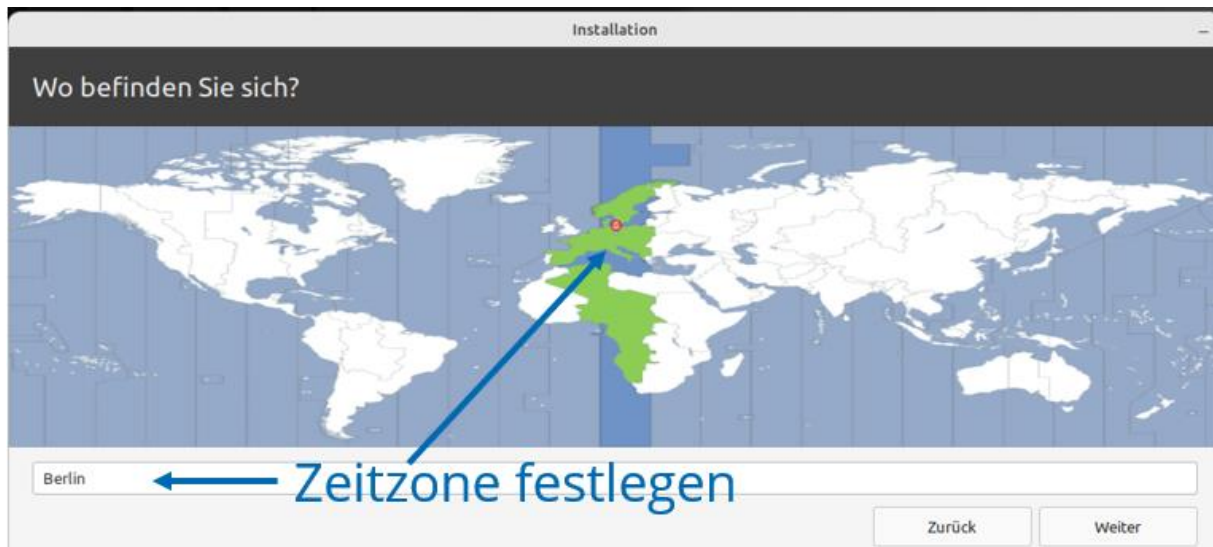


Nach dem Klick auf den Button „Jetzt installieren“ öffnet sich ein Fenster, wobei man die Partitionierung und Formatierung der Festplatte bestätigen muss.

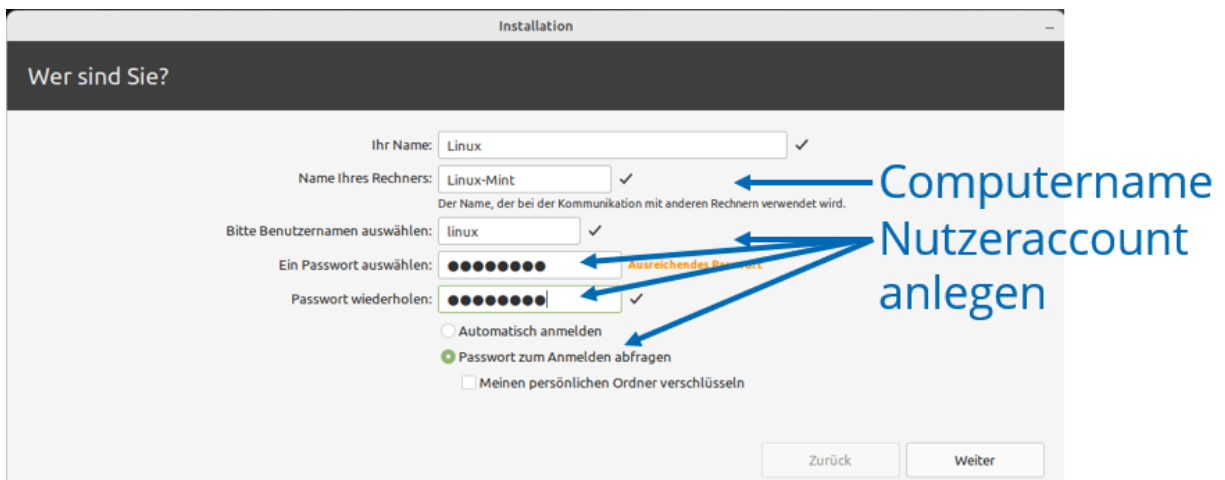




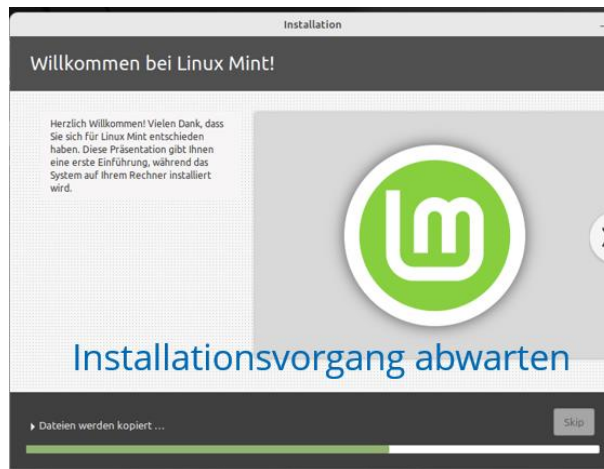
Im nächsten Schritt legt man die Zeitzone fest.



Über das Fenster „Wer sind Sie?“ wählt man den Computernamen und legt einen Nutzeraccount an.



Nachdem man diese Werte festgelegt hat, startet der Installationsvorgang.



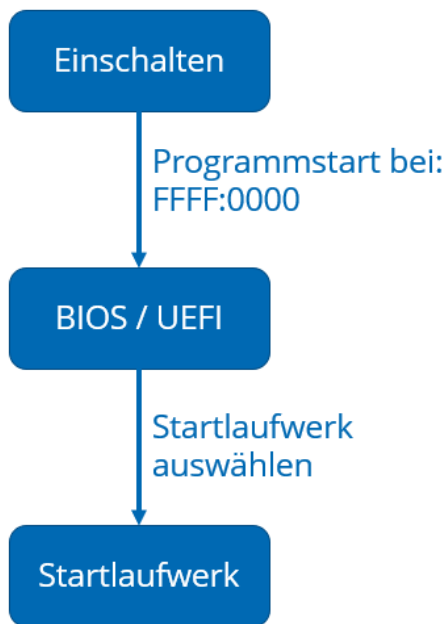
Wenn die Installation abgeschlossen ist, öffnet sich ein neues Fenster. Hier wählt man den Button „Jetzt neu starten“ aus, um einen Neustart des Systems durchzuführen.



Beim Neustart startet man Linux Mint nun von der Festplatte und nicht mehr vom Live-Medium.



## 1.2.2 Bootvorgang



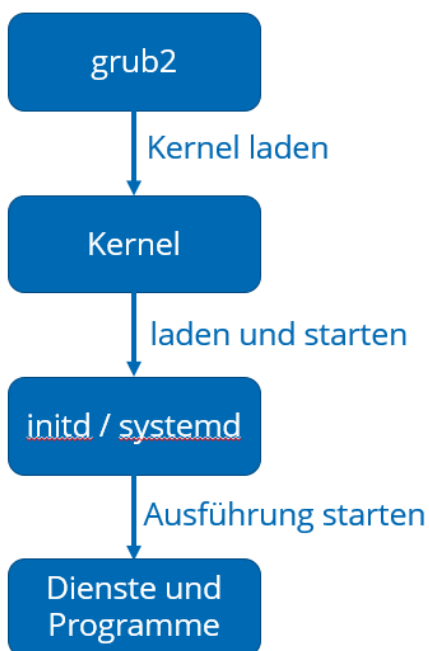
Der Startprozess des Computers besteht aus mehreren Stufen. Beginnend erfolgt das Einschalten. Hierbei werden die PWR\_SW-Pins kurzgeschlossen und so die Hardware mit Strom versorgt. Der Programmstart beginnt bei FFFF:0000.

Als nächstes startet das BIOS. Hier wird ein Kalt- oder ein Warmstart (Adresse 0000:0472 = 1234) durchgeführt. Weiterhin erfolgt hier eine Systemdiagnose (Power on Self Test = POST) und es wird auf das Startlaufwerk (HDD, SSD, USB oder DVD) zugegriffen.

Beim ausgewählten Startlaufwerk wird der Master Boot Record (MBR) ausgeführt. Dabei wird die Partitionstabelle ausgewertet, die primäre Partition geladen und das Ganze an den Boot Manager (NTLDR/ BOOTMGR / grub2) übergeben.

Beim Start von UEFI wird der Boot Manager direkt geladen und es erfolgt die Übergabe an den Boot Manager.

### 1.2.2.1 Bootvorgang Linux



Bei Linux wird als Boot Manager der Grand Unified Bootloader 2 (GNU) eingesetzt. Dieser wählt das Kernel Image aus. Die Konfiguration des Boot Managers befindet sich unter /boot/grub/grub.conf oder /etc/grub.conf. Der Boot Manager mounted (einbinden) den Kernel im File System.

Der Kernel wird von /sbin/init (PID 1) gestartet. Dies erfolgt mithilfe des temporären Dateisystems Initial RAM Disk (initrd). Initrd wird ausgewertet und gestartet.

Initrd wertet im Anschluss Run-Level aus (/etc/rc0-6.de) und systemd startet die Programme (/etc/init.d/<Programm>). Diese Prozesse starten die Ausführung von Diensten und Programmen, wobei Programme ihren eigenen Start verwalten.

### 1.2.2.2 Run-Level

Bei Linux existieren die folgenden sechs Run-Level:

- 0 - System halt
- 1 - Single user mode (als root)
- 2 - Multiple user mode, ohne Netzwerk, mit GUI
- 3 - Multiple user mode, mit Netzwerk, ohne GUI
- 4 - User-definable
- 5 - Multiple user mode, mit Netzwerk, mit GUI (Standard)
- 6 - Reboot

### 1.2.3 FHS - Verzeichnisstruktur unter Linux

Die allgemeine Annahme beim Filesystem Hierarchy Standard ist, dass alles eine Datei ist. Prozesse, der Arbeitsspeicher, die Festplatte, eine Partition, ein Netzwerkadapter, eine Textausgabe auf der Konsole und eine Datei – dies alles sind Dateien.

Unter Linux/BSD gibt es einige Grundregeln. Es gibt nur eine einzige Verzeichnisstruktur. Außerdem ist alles eine Datei. Die letzte Grundregel ist, dass Dateien in der Verzeichnisstruktur eingebunden sind. Den Standard kann man unter <https://refspecs.linuxfoundation.org/fhs> abrufen.

Die folgende Abbildung zeigt den FHS, indem im Root Verzeichnis mit dem Befehl `ls -la` alle Dateien und Verzeichnisse angezeigt werden.

```
linux@Linux-Mint:/$ ls -la
insgesamt 80
drwxr-xr-x 19 root root 4096 Apr 19 09:26 .
drwxr-xr-x 19 root root 4096 Apr 19 09:26 ..
lrwxrwxrwx 1 root root 7 Apr 19 09:17 bin -> usr/bin
drwxr-xr-x 4 root root 4096 Apr 19 09:35 boot
drwxr-xr-x 2 root root 4096 Apr 19 09:26 cdrom
drwxr-xr-x 20 root root 4180 Apr 19 09:55 dev
drwxr-xr-x 144 root root 12288 Apr 19 09:38 etc
drwxr-xr-x 3 root root 4096 Apr 19 09:26 home
lrwxrwxrwx 1 root root 7 Apr 19 09:17 lib -> usr/lib
lrwxrwxrwx 1 root root 9 Apr 19 09:17 lib32 -> usr/lib32
lrwxrwxrwx 1 root root 9 Apr 19 09:17 lib64 -> usr/lib64
lrwxrwxrwx 1 root root 10 Apr 19 09:17 libx32 -> usr/libx32
drwx----- 2 root root 16384 Apr 19 09:16 lost+found
drwxr-xr-x 2 root root 4096 Jan 4 15:14 media
drwxr-xr-x 2 root root 4096 Jan 4 15:14 mnt
drwxr-xr-x 2 root root 4096 Jan 4 15:14 opt
dr-xr-xr-x 251 root root 0 Apr 19 09:54 proc
drwx----- 5 root root 4096 Apr 19 10:09 root
drwxr-xr-x 34 root root 940 Apr 19 10:09 run
lrwxrwxrwx 1 root root 8 Apr 19 09:17 sbin -> usr/sbin
drwxr-xr-x 2 root root 4096 Jan 4 15:14 srv
dr-xr-xr-x 13 root root 0 Apr 19 09:54 sys
drwxrwxrwt 17 root root 4096 Apr 19 10:09 tmp
drwxr-xr-x 14 root root 4096 Jan 4 15:14 usr
drwxr-xr-x 11 root root 4096 Jan 4 15:48 var
```

/ ist das root- oder Wurzelverzeichnis.

. Der einzelne Punkt stellt das aktuelle Verzeichnis dar (current folder). Dieser wird in jedem Verzeichnis aufgelistet.

.. Die zwei Punkte stehen für das übergeordnete Verzeichnis (parent folder). Auch das übergeordnete Verzeichnis wird in jedem Verzeichnis aufgelistet.

/dev steht für devices (Geräte). Es dient der direkten Einbindung von Hardware. Es beinhaltet kein Filesystem, sondern nur Binärwerte.

- psaux: PS-Maus / Tastatur
- sd: SATA-Festplatte (sdb3 → 2. Festplatte, 3. Partition)
- nvme: NVME-SSD (nvme5n1p4 → 6. Festplatte, 4. Partition)
- tty: Serielle Terminals (ttyUSB0 → USB-Gerät)
- vcsu: Scrollbuffer für serielle Terminals
- random: Zufallszahlen
- null: lesen immer 0, schreiben wird verworfen

In /etc befinden sich die globalen Konfigurationen:

- /etc/crontab: Zeitgesteuerte Ausführungen (wöchentliches Backup)
- /etc/cron.X: Ordnerstruktur von crontab (stündlich, täglich, wöchentlich, monatlich)
- /etc/fstab: Festplatteneinbindung (Mountpoints)
- /etc/hosts: statische DNS-Auflösung
- /etc/shadow: Benutzer, Passwort-Hash, Passwortwechsel-Zeitstempel
- /etc/ssh: SSH-Konfiguration

Außerdem gibt es hier Standarddateien für benutzerspezifische Einstellungen, welche auch als Skeleton-Dateien bezeichnet werden. Dazu zählt beispielsweise /etc/profile, welches zu /home/Username/.profile wird. Weitere Skeleton Dateien finden sich unter /etc/skel/.

/home enthält die Unterverzeichnisse der Benutzer. Der eigene Unterordnerpfad ist auch als “~” aufrufbar, welches /home/myfolder entsprechen würde. Das Home-Verzeichnis ist der Speicherort für persönliche Dateien (Bilder, Downloads und Dokumente), die individuellen Konfigurationen für Programme und SSH-Keys (/home/user/.ssh). Benutzer besitzen jedoch nicht zwangsläufig ein Home-Verzeichnis.

/root ist das home-Verzeichnis des root-Nutzeraccounts. Es besitzt den gleichen Aufbau wie das home-Verzeichnis von anderen Nutzern.

/lib enthält Softwarebibliotheken, sprich eine Sammlung von Programmfunktionen. Diese sind Prozessorunspezifisch. /lib32 und /libx32 enthalten Bibliotheken für die x86-Architektur, während /lib64 Bibliotheken für die x64-Architektur enthält.

In /media sind externe Speichergeräte wie CDs, DVDs, Festplatten oder Ähnliches aufgelistet.

/mnt steht für mount und enthält die eingebundenen Dateisysteme.

/opt steht für optional und enthält Zusatzpakete und Add-ons (jre, jdk, ...).

/proc steht für processes. Dieses Verzeichnis enthält Prozessinformationen und kann nur lesend aufgerufen werden.

/sys steht für system-processes. Auch dieses Verzeichnis ist nur lesbar und enthält Prozessinformationen zu Systemprozessen, sprich Kernel-Prozesse.

/srv steht für serve. Hier befinden sich globale Dateien für den Netzwerkshare. Beispielsweise sind unter /srv/www globale Dateien für Webserver, unter /srv/ftp globale Dateien für den FTP-Server und unter /srv/samba globale Dateien für diesen FTP-Server.

/usr steht für user und enthält die Benutzerdateien für den Netzwerkshare und ist auch für andere Benutzer nutzbar. Bin enthält ausführbare Programme (Benutzerbefehle), include enthält header-Dateien (C und C++ Programmierung), lib enthält Bibliotheken und sbin enthält OS-unabhängige Programme.

/tmp steht für temporary data und enthält temporäre Dateien sowie den Cache.

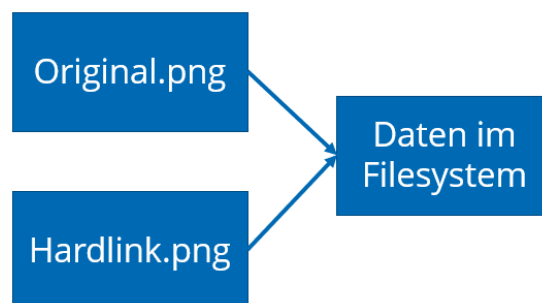
/var steht für variable Data und beinhaltet häufige Veränderungen. Dazu zählen der Cache (/var/cache), Logs (/var/log), Webseiten (var/www), temporäre Dateien (/var/tmp), Mails, unter anderem auch die systeminternen Mails an den Administrator (/var/mail) und Spools wie die Druckerschlange (/var/spool).

Der FHS kann als Good-Practise angesehen werden und Distributionen können abweichen. Beispielsweise

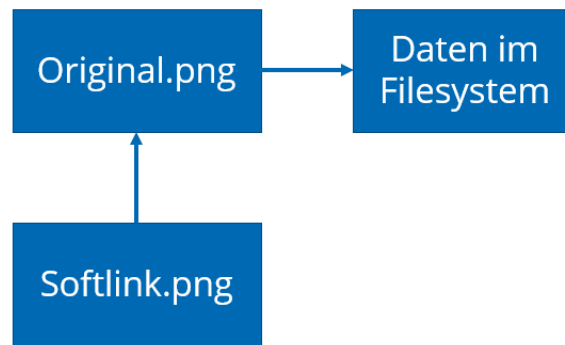
- /usr/bin, /bin, /sbin
- /var/www und /srv/www
- /mnt und /media
- /boot/grub/grub.conf und /etc/grub.conf

#### 1.2.4 Verlinkung von Dateien

Hardlinks stellen Verlinkungen zu gleichen Dateien dar und sind somit ein direkter zweiter Verweis. Durch Löschen der Originaldatei verschwindet nur die Verknüpfung der Original-Datei. Die Daten sind jedoch weiter über den Hardlink erreichbar. Wird der Hardlink gelöscht, wird auch die Verknüpfung des Hardlinks gelöscht. Die Dateien sind jedoch weiterhin über das Original erreichbar.



Der Softlink, auch Symbolic-Link genannt, ist eine Verlinkung zu einem Dateiverweis. Somit stellt der Softlink einen indirekten Verweis dar. Wird die Original-Datei gelöscht, sind die Daten komplett gelöscht und der Softlink führt ins Leere. Das Löschen des Softlinks bewirkt das Löschen der Verknüpfung des Softlinks, die Daten sind jedoch weiterhin über das Original erreichbar.



### 1.2.5 Zusammenfassung

In der heutigen Präsentation haben Sie gelernt, welche Auswahlkriterien zu treffen sind, um ein passendes Installationsimage für eine Installation auszuwählen.

Am Beispiel von Linux Mint lernten Sie den Installationsablauf.

Des Weiteren ist Ihnen nun der Filesystem Hierarchy Standard bekannt. Insbesondere kennen Sie, welche Daten in den entsprechenden Unterordnern hinterlegt werden.

Abschließend lernten Sie die zwei verschiedenen Verlinkungsarten Hard- und Softlink unter Linux kennen.

## 1.3 Linux Grundlagen II

### 1.3.1 Netzwerk-Konfiguration

#### 1.3.1.1 Interface-Konfiguration

Die Interface-Konfiguration für Debian/Ubuntu befindet sich unter `/etc/network/interfaces`. Die statische Konfigurierung ist:

```
auto eth0
iface eth0 inet static
    address 192.168.0.100
    netmask 255.255.255.0
    gateway 192.168.0.1
```

`auto eth0` steht dafür, dass automatisch ein interface up bei einer physischen Verbindung durchgeführt wird. Mithilfe von `iface eth0 inet static` wird das Interface `eth0` statisch mit IPv4 konfiguriert. `address` zeigt die zugewiesene IP-Adresse an, `netmask` die Subnetzmaske (Netzwerkgröße) und `gateway` den Weiterleitungsknoten für externe Netzwerke.

Die dynamische Konfigurierung erfolgt wie folgt:

```
auto eth0
iface eth0 inet6 dhcp
```

`iface eth0 inet6 dhcp` bedeutet, dass die Interface `eth0` IPv6-Konfiguration über DHCP durchgeführt wird. Die DHCP-Optionen liefern den DNS, etc.

Um die IP-Konfiguration auszulesen, kann man folgende Befehle verwenden:

- `ifconfig -a`
- `netstat -ei`
- `ip address`

Um aktive Verbindungen und lauschende Ports anzeigen zu lassen, verwendet man den Befehl `netstat -tulpen`. Für das Auslesen statischer Routen verwendet man `route`.

Die Interface-Konfiguration für SUSE befindet sich unter `/etc/sysconfig/network/if`. In diesem Fall sieht die statische Konfigurierung wie folgt aus:

```
BOOTPROTO='static'
IPADDR='192.168.0.100'
MTU='1500'
NAME=''
NETMASK='255.255.255.0'
STARTMODE='auto'
USERCONTROL='no'
```



Jedes Interface besitzt eine eigene Datei, wobei der Dateiname interfacename ist. BOOTPROTO='static' bedeutet, dass die IP statisch ist. MTU='1500' legt die Maximal Transmission Unit, also die Nutzlastgröße, fest. STARTMODE='auto' sorgt dafür, dass das interface up automatisch bei einer physischen Verbindung erfolgt. USERCONTROL='no' legt fest, dass eine Einstellung über die GUI nicht möglich ist.

### 1.3.1.2 WLAN-Netzwerke

Über den WLAN-Manager erfolgt die Konfiguration für den Login, wobei das Passwort im Klartext vorliegt. Unter Ubuntu befindet sich die Konfiguration unter /etc/**NetworkManager**/**<WLAN-SSID>**, bei Debian/SUSE unter /etc/**wpa\_supplicant**/wpa\_supplicant.conf und für Red Hat/Fedora unter /etc/**NetworkManager**/**<WLAN-SSID>**, wobei hier die Konfiguration über das eigene Tool nmcli durchgeführt wird.

### 1.3.2 Dateisysteme Überblick

Die folgende Abbildung zeigt die von gdisk unterstützten Filesysteme. Es ist ersichtlich, dass Linux fast alle Dateisysteme unterstützt.

Type search string, or <Enter> to show all codes:	a023 Android MSDP	a024 Android DPO	b19a Solaris Reserved 4	b19b Solaris Reserved 3
0780 Microsoft basic data	0c01 Microsoft reserved	a025 Android recovery 2	c001 HP-LUX data	c002 HP-LUX service
1700 Windows RE	3000 OEM boot	a026 Android persist	e100 ONE boot	e101 ONE config
3003 SMI config	3900 Plan 9	a027 Android modem ST1	e400 Veracrypt data	e401 Veracrypt sboot
4100 PowerPC PREP boot	4200 Windows LDM data	a029 Android FSC	e500 NixOS SFS	e501 Sony system partition
4201 Windows LDM metadata	4202 Windows Storage Spaces	a02b Android FSC 2	e601 Lenovo system partition	e700 EFI system partition
7501 IBM GPFS	7100 ChromeOS kernel	Press the <Enter> key to see more codes, q to quit:	e702 BIOS boot partition	e703 BIOS boot partition
7f01 ChromeOS root	7f02 ChromeOS reserved	a02d Android keystore	f800 Ceph OSD	f801 Ceph dm-crypt OSD
9200 Linux swap	9300 Linux filesystem	a02f Android ENKST	f802 Ceph Journal	f803 Ceph dm-crypt Journal
9301 Linux reserved	9302 Linux /home	a031 Android spare1	Press the <Enter> key to see more codes, q to quit:	f804 Ceph disk in creation
9303 Linux x86 root (/)	9304 Linux x86-64 root (/)	a033 Android spare3	f805 Ceph block	f806 Ceph dm-crypt disk in creation
9305 Linux ARMv4 root (/)	9306 Linux /srv	a035 Android raw resources	f807 Ceph block DB	f808 Ceph lockbox for dm-crypt keys
9307 Linux ARMv2 root (/)	9308 Linux dm-crypt	a037 Android FOTA	f809 Ceph multipath OSD	f80a Ceph multipath journal
9309 Linux LUKS	930a Linux IA-64 root (/)	a039 Android cache	f80c Ceph multipath block 1	f80d Ceph multipath block 2
930b Linux x86 root verity	930c Linux x86-64 root verity	a03b Lg (Android) advanced flasher	f80e Ceph multipath block DB	f80f Ceph multipath block write-ahead
930d Linux ARMv2 root verity	930e Linux ARMv4 root verity	a03d Android PG2FS	f810 Ceph dm-crypt block	f811 Ceph dm-crypt block DB
930f Linux IA-64 root verity	9310 Linux /var	a03f Android MFG	f812 Ceph dm-crypt block write-ahead lo	f813 Ceph dm-crypt LUKS Journal
9311 Linux /var/tmp	9400 Intel Rapid Start	a008 Atari TOC basic data	f814 Ceph dm-crypt LUKS block	f815 Ceph dm-crypt LUKS block DB
9500 Container Linux /usr	9501 Container Linux resizable roots	a503 FreeBSD UFS	f816 Ceph dm-crypt LUKS block write-ah	f817 Ceph dm-crypt LUKS OSD
9502 Container Linux /OEM customization	9503 Container Linux root on RAID	a505 FreeBSD Vium/RAID	f808 VMware WFS	f809 VMware reserved
9600 Linux LVM	a000 Android bootloader	a507 Midnight BSD UFS	f800 VMware kcore crash protection	f800 Linux RAID
a001 Android bootloader 2	a002 Android boot 1	a501 FreeBSD boot		
Press the <Enter> key to see more codes, q to quit:	a004 Android misc	a508 Midnight BSD data		
a003 Android recovery 1	a006 Android system 1	a581 Midnight BSD swap		
a005 Android metadata	a008 Android data	a584 Midnight BSD ZFS		
a007 Android cache	a009 Android factory	a585 Midnight BSD ZFS		
a009 Android persistent	a00a Android OEM	a600 OpenBSD disklabel		
a00b Android fastboot/tertiary	a00c Android config	a901 NetBSD swap		
a00e Android vendor	a010 Android meta	a903 NetBSD UFS		
a00f Android factory (alt)	a012 Android SBL1	a904 NetBSD concatenated		
a011 Android EXT	a014 Android SBL2	a906 NetBSD RAID		
a013 Android SBL2	a016 Android QSEE/tz	Press the <Enter> key to see more codes, q to quit:		
a015 Android APPSBL	a018 Android RPM	a100 Apple HFS/HFS+		
a017 Android QREE/hyp	a01a Android DDR	a102 Apple RAID offline		
a019 Android WDOG debug/sdi	a01c Android RAM dump	a104 AppleTV recovery		
a01b Android CDT	a01e Android PMIC	a106 Apple SoftRAID Status		
a01d Android SEC	a020 Android misc 2	a108 Apple SoftRAID Volume		
a01f Android misc 1	a022 Android APDP	a10a Apple APFS		
a021 Android device info		bc00 Acronis Secure Zone		
		b100 Solaris root		
		b101 Solaris /usr & Mac ZFS		
		b102 Solaris swap		
		b104 Solaris /var		
		b106 Solaris alternate sector		
		b108 Solaris Reserved 2		
		b109 Solaris Reserved 3		

Die wichtigsten Dateisysteme sind im Folgenden dargestellt:

Dateisystem	technischer Stand	Von Betriebssystem unterstützt	Rechteverwaltung	Journaling
<b>Btrfs</b>	Standard in Suse Linux, optional bei allen Distributionen seit 2014	Linux, ReactOS	✓	🚫 Nur Metadaten
...2	Ehemaliges Standarddateisystem, sehr ausgereift, aber aufgrund fehlender Journal-Funktion sehr langsam bei den Dateisystem-Checks	Linux; BSD; Windows mit  ext2fd  oder ExtFS für Windows ; MacOS X mit ExtFS-Treiber	✓	✗
<b>ext...</b>	...3 Ehemaliges Standarddateisystem	Linux; BSD; Windows mit  ext2fsd  oder ExtFS für Windows ; MacOS X mit ExtFS-Treiber	✓	✓
...4	Aktuelles Standarddateisystem für Ubuntu	Linux; Windows mit  ext2fd  oder ExtFS für Windows ; MacOS X mit ExtFS-Treiber	✓	✓
<b>JFS</b>	Kompromiss aus Schnelligkeit und Sicherheit	Linux; Unix; OS/2	✓	🚫 Nur Metadaten
<b>ReiserFS</b>	war lange Standarddateisystem vieler Distributionen	Linux; BSD; Windows mit kommerz. Zusatztreiber	✓	✓
<b>Reiser4</b>	experimentell, schon sehr lange in der Entwicklung	Linux (nur mit Kernelpatch)	✓	✓
<b>XFS</b>	ausgereift und stabil, im Desktop-Bereich aber nicht sehr verbreitet	Linux; Unix; BSD	✓	✓
...12			✗	✗
<b>FAT...</b>	...16 Dateisysteme von Microsoft, als Installationsdateisystem für Linux nicht geeignet	Windows; Linux; BSD; Mac OS	✗	✗
...32			✗	✗
<b>exFAT</b>	proprietär, speziell für  Flash-Speicher, als Installationsdateisystem für Linux nicht geeignet	Windows; Linux, FreeBSD via FUSE-Treiber; Mac OS	✗	✗
<b>NTFS</b>	das Standarddateisystem von Windows XP, Vista, 7 und 8, als Installationsdateisystem für Linux nicht geeignet	Windows; Linux; BSD; Mac OS	in Linux nur bedingt verwendbar*	🚫 Nur Metadaten
<b>ZFS</b>	Standarddateisystem unter Solaris	Solaris; Linux; BSD	✓	✓

### 1.3.2.1 Software-RAIDs

RAID steht für Redundant Array of Independent Disks und bezeichnet die Zusammenschaltung von einzelnen Datenträgern zu einer virtuellen Festplatte. Software-RAIDs haben keinen Hardware-Controller. Hier übernimmt die CPU die Berechnungen und der Cache von Festplatten ist deaktiviert, da sonst ein Datenverlust möglich wäre. Der Superblock ist für die Speicherung der Meta-Informationen über das RAID verantwortlich.

MDADM steht für Multiple Disk ADMINistration. Dieses Linux RAID (0xFD MBR, 0xFD00 GPT) ist als Filesystem auf der Partition. Bei Version 1.0 ist der Superblock am Ende der Partition, wohingegen er sich bei Version 1.1 am Anfang befindet. Bei der Version 1.2 ist der Superblock 4KB nach dem Anfang der Partition.

LVM steht für Logical Volume Manager, welcher dynamisch veränderbare Partitionen (auch über mehrere Datenträger) enthält. LVM ist in grub2 (Bootmanager) integriert.

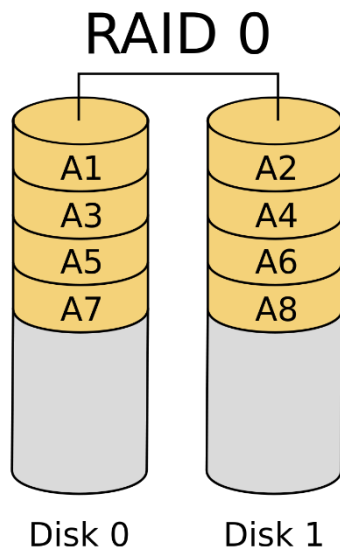
ZFS steht für Zettabyte File System. Dieses Software RAID ist im Filesystem integriert und besitzt Pools für die Speicheraufteilung. Die Fehlererkennung erfolgt über Merkle-Trees (Hash-Baum).

### 1.3.2.2 RAID-Varianten

Es gibt verschiedene Varianten von RAID-Systemen. JBOD steht für Just a Bunch Of Disks und bezeichnet eine Aneinanderkettung von Datenträgern. RAID-0 (Stripe) Systeme verteilen die Speicherblöcke, wohingegen RAID-1 (Mirror) Systeme die Speicherblöcke spiegeln. In einem RAID-3 werden die Bytes verteilt und jeweils Prüfsummen auf einem dedizierten Datenträger gespeichert. RAID-4 Systeme verteilen die Speicherblöcke und haben ebenfalls die Prüfsumme auf einem dedizierten Datenträger.

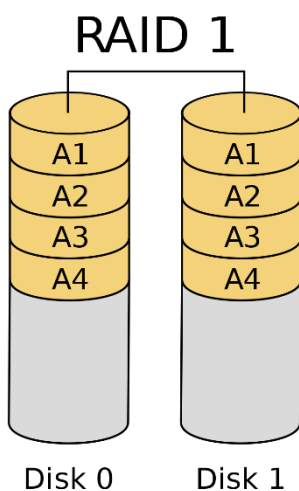
Die RAID-5 bzw. Parity Systeme verteilen ebenfalls die Speicherblöcke und auch hier gibt es Prüfsummen, welche aber nicht auf einem dedizierten Datenträger abgelegt sind. RAID-6 Systeme verteilen die Speicherblöcke und bilden zwei Prüfsummen. RAID-10 Systeme sind eine Kombination aus zwei RAID-1 Systemen, welche mit einem RAID-0 zusammengeführt werden.

#### RAID-0



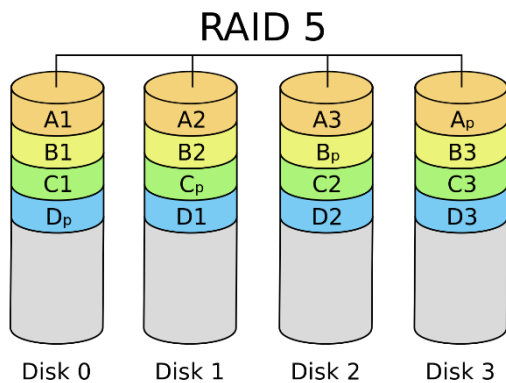
Bei einem RAID-0 System werden die Daten in Blöcke zerlegt. Diese werden wie Spielkarten reihum an die Festplatten verteilt. Bei zwei Datenträgern hat man somit die doppelte Schreibgeschwindigkeit, die doppelte Lesegeschwindigkeit, die doppelte Ausfallwahrscheinlichkeit und 0% Speicherplatzverlust.

#### RAID-1



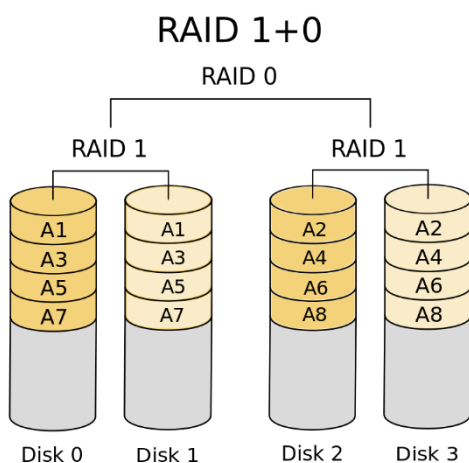
Bei RAID-1 Systemen werden die Daten in Blöcke zerlegt und die Blöcke auf allen Datenträgern gespeichert. Bei zwei Datenträgern hat man somit die halbe Schreibgeschwindigkeit, die doppelte Lesegeschwindigkeit, die halbe Ausfallwahrscheinlichkeit und 50% Speicherplatzverlust.

## RAID-5



Bei einem RAID-5 werden die Daten in Blöcke zerlegt und wie schon beim RAID-0 wie Spielkarten reihum an die Festplatten verteilt. Die Prüfsumme wird in einem zusätzlichen Block gespeichert. Bei drei Datenträgern ergeben sich folgende Werte: fast doppelte Schreibgeschwindigkeit, doppelte Lesegeschwindigkeit, verringerte Ausfallwahrscheinlichkeit und 33% Speicherplatzverlust.

## RAID-10



RAID-10 kombiniert die Vorteile des RAID-0 und des RAID-1 Systems. Die Daten werden auch hier in Blöcke zerlegt. Bei vier Datenträgern erhält man fast die doppelte Schreibgeschwindigkeit, die doppelte Lesegeschwindigkeit, eine verringerte Ausfallwahrscheinlichkeit und 50% Speicherplatzverlust.

### 1.3.2.3 /etc/fstab

Die fstab-Datei beschreibt die Dateisysteme und dessen Mount-Points. Die Einbindung erfolgt nach einem Neustart und nach einer neuen Verbindung im Betrieb (USB-Stick, externe Festplatte, Netzwerkshare). Die Einträge der fstab-Datei sind zeilenweise angelegt. Dabei werden diese manuell hinzugefügt und besitzen eine Kennung (Devicepfad, UUID). Tabs oder Leerzeichen werden für die Trennung eingesetzt und „#“ kennzeichnet einen Kommentar. Leerzeilen werden bei der Auswertung ignoriert.

Ein Beispieleintrag der fstab-Datei wäre: **UUID=03b77228-ed4c-4218-910e-11b9f77c4b46 /backup ext4 defaults 0 2**

- **UUID=03b77228-ed4c-4218-910e-11b9f77c4b46** ist die eindeutige Hardwarekennung. Alternativ könnte man auch /dev/sda1 schreiben, was jedoch abhängig von der Verbindungsreihenfolge ist

- **/backup** kennzeichnet den Mountpoint (Einbindung ins FHS). None steht für Swap (Auslagerungsspeicher, ein Leerzeichen bedeutet \040 und ein Tab steht für \011
- **ext4** kennzeichnet den Filesystemtyp
- **defaults** sind Optionen, welche nur durch Kommata getrennt werden. defaults = rw,suid,dev,exec,auto, nouser,async
- **0** beantwortet die Frage, ob ein Backup mit dump durchgeführt werden soll. 0 steht für nein und 1 für ja
- **2** kennzeichnet die Überprüfungsreihenfolge beim Boot-Vorgang. 0 steht für keine Überprüfung, 1 steht für zuerst (meist Root-FS) und 2 für anschließend (meist zusätzliche Datenträger)

#### 1.3.2.4 Partitionieren

Für das Partitionieren stehen zwei Kommandozeilenprogramme zur Verfügung. fdisk kennt die Partitionierung mit MBR (Master Boot Record), was auch als DOS bezeichnet wird, mit GPT (Globally Unique Identifier Partitioning Table), mit SGI (Silicon Graphics and Irix Partitionierung) und mit Sun (Sun Systems Partitionierung).

gdisk kennt Partitionierungen mit MBR (Master Boot Record), mit BSD (auch BSD-Disklabel), mit APM (Apple Partitioning Map) und mit GPT (Globally Unique Identifier Partitioning Table).

#### 1.3.2.5 Formatieren

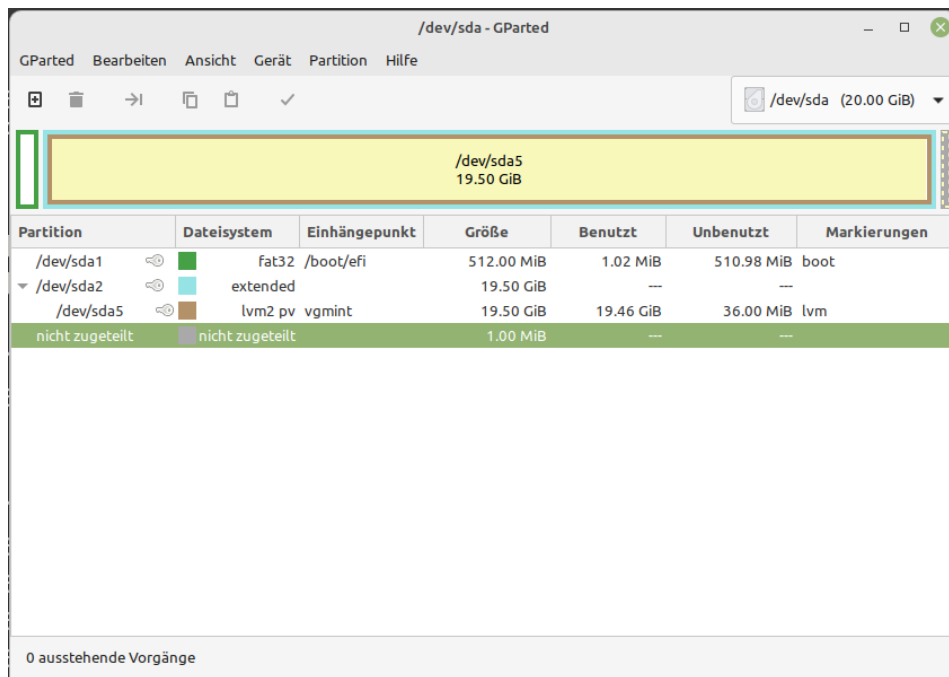
Das Formatieren wird mit mkfs durchgeführt, welches für make filesystem steht. mkfs ist ein Kommandozeilenprogramm, welches Partitionen formatiert. Filesystem spezifische Unterprogramme sind

- mkfs.bfs
- mkfs.ext2
- mkfs.ext3
- mkfs.ext4
- mkfs.msdos
- mkfs.minix
- mkfs.vfat

Der Befehl `mkfs -t ext4 /dev/sdb1` spricht auf der zweiten Festplatte die erste Partition an, wobei die Partitionierung mit ext4 erfolgt. Der Befehl `mkfs -t ntfs /dev/sdc4` bezieht sich auf die vierte Partition der dritten Festplatte und die Partitionierung erfolgt mit ntfs.

#### 1.3.2.6 Partitionieren und Formatieren

GParted steht für Gnome Partition Editor und stellt eine Graphische Oberfläche auf libparted dar. GParted wird von vielen Distros verwendet und ist auch als eigene Distro (Live Version) erhältlich. Es kennt fast alle Partitionierungen und unterstützt viele Filesysteme. Weiterhin ist es ähnlich zur Windows Datenträgerverwaltung.



### 1.3.3 Benutzer und Gruppen

#### 1.3.3.1 Speicherorte Benutzer

Informationen zu den Benutzern werden an verschiedensten Stellen gespeichert. Die Datei `/etc/passwd` ist von allen Nutzern lesbar und beinhaltet eine Auflistung aller Nutzer. Früher enthielt sie ebenfalls einen Hash. Die Datei `/etc/shadow` ist der Speicherort des Passwort-Hashes, welche nur vom Root-Nutzer lesbar ist. In der Datei `/etc/group` befindet sich eine Auflistung von Gruppen und deren Nutzern. In den Verzeichnissen `/home` oder `/root` befinden sich die privaten Dateien von Nutzern.

#### `/etc/passwd`

In der Datei `/etc/passwd` bildet der `:` (Doppelpunkt) das Trennzeichen. Ein Eintrag setzt sich aus folgenden Angaben zusammen: Benutzername, Passwort (x bedeutet, dass das Passwort in `/etc/shadow` steht), Benutzer-ID, Gruppen-ID, GECOS (Zusatzinformationen wie E-Mail, Telefonnummer oder Kommentare, welche durch Kommata getrennt sind), das Homeverzeichnis und die Standard-Shell. Ein Beispieleintrag könnte wie folgt aussehen: **Hans:x:0:0:hans@hsmw.de:/home/hans:/bin/bash**

```

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin

```

## [/etc/shadow](#)

Wie auch in der Datei `/etc/passwd` ist der `:` (Doppelpunkt) hier das Trennzeichen. Ein Eintrag der Datei ist aus folgenden Informationen zusammengesetzt:

- Benutzername
- Passwort (`$Type$Salt$Hash`)
  - `$1`: MD5
  - `$2a`: Blowfish
  - `$2y`: Eksblowfish
  - `$5`: SHA-256
  - `$6`: SHA-512
  - `!`, `!!`, `*` für kein Login möglich (meist bei `/bin/false` als Shell)
- Letzte Passwortänderung in Unix-Time
- Minimales Passwortalter
- Maximales Passwortalter
- Warnzeitraum bis ungültig
- Inaktivitätszeitraum
  - Zeit zwischen Passwort zu alt und Login verwehrt
- Deaktiviert Zeitstempel
- Unbenutzt / reserviert

## [/etc/group](#)

Wie auch in der Datei `/etc/passwd` ist der `:` (Doppelpunkt) hier das Trennzeichen. Die Einträge bestehen aus dem Gruppennamen, dem Passwort, der Gruppen-ID und der Mitgliederliste, bei welcher die Informationen durch Kommata getrennt sind. Ein Eintrag der `/etc/group` Datei könnte wie folgt aussehen: **`sudo:x:27:Hans,Peter`**

Jeder Nutzer ist mindestens in einer Gruppe, wobei der Gruppenname standardmäßig dem Benutzernamen gleich ist. Weiterhin besitzen einige Distros Backups der `passwd`, `shadow` und `group` Dateien. Hierbei wird das Backup automatisch bei einer Änderung erstellt.

[/home oder /root](#)

Unter `/home` sind die Dateien und Verzeichnisse für den `root`-Nutzer, unter `/home` die für alle anderen Nutzer. `/home` kann manchmal auch weiter in einzelne Abteilungen gegliedert sein. Nicht jeder Benutzer besitzt ein Home-Verzeichnis (`www-data` und `sshd`). In den Verzeichnissen kann es versteckte Ordner geben, welche mit einem `.` als erstes Zeichen gekennzeichnet sind. Diese enthalten Konfigurationen und jedes Programm besitzt einen eigenen Unterordner. Der `.cache` Ordner ist beispielsweise ein temporärer Speicher von Anwendungen. Die Standardordner bei Desktop-Linux sind:

- Bilder (Pictures)
- Dokumente (Documents)
- Downloads (Downloads)
- Musik (Music)
- Öffentlich (Public)
- Schreibtisch (Desktop)
- Videos (Videos)
- Vorlagen (Templates)

### [1.3.3.2 Benutzerrechte](#)

Da bei FHS die Ansicht vertreten wird, dass alles eine Datei ist, werden die Benutzerrechte über Datei-Zugriffsrechte realisiert. Somit bedeuten Leserechte auf Dateien anderer Nutzer, dass man Leserechte für die Verzeichnisse `/home/userX` hat.

Wenn man Schreibrechte auf Festplatte 1 hat, bedeutet dies, dass man Schreibrechte auf `/dev/sda` hat. In Bezug auf Webserver-Dateien hat man Schreibrechte im Ordner `/var/www` und in Bezug auf Passwortänderungen hat man Schreibrechte auf die `/etc/shadow` Datei.

Hat der Nutzer Ausführrechte, um den Computer auszuschalten, bedeutet dies, dass er Ausführrechte auf `/sbin/shutdown` hat.

### [1.3.3.3 Datei-Zugriffsberechtigung](#)

Datei-Zugriffsberechtigungen besitzen folgendes Format: `drwxrwxrwx` (TypBesitzerGruppeAlleAnderen). Die eingesetzten Zeichen bedeuten:

- d: Dateiformat
- r: Leserechte (read)
- w: Schreibrechte (write)
- x: Ausführrechte (execute)
- s: setuid (Ausführung mit Besitzer-rechten)
- S: setgid (Ausführung mit Gruppen-rechten)
- t: stickybit (Proramm verbleibt im Speicher nach Ausführungsende)



Als Dateiformat sind folgende Eintragungen möglich:

- -: normale Datei
- d: Ordner (directory)
- b: Block-Device
- c: Zeichen (character)
- l: Softlink (symbolic link)
- n: Netzwerk (network)
- p: Pipe / First-in, First-out
- s: Socket

Diese Datei-Zugriffsberechtigungen sollen an einem Beispiel erläutert werden. Der Befehl `ls -l /dev/sda` erzeugt folgende Ausgabe: `brw-rw---- 1 root disk 8, 0 Apr 20 11:42 /dev/sda`

In diesem Fall ist der Besitzer `root` und die Gruppe `disk`. `b` steht für block device (Speichermedium). Laut den Zugriffsberechtigungen darf `root` lesen und schreiben, aber nicht ausführen. Auch die **Gruppenmitglieder von disk dürfen lesen und schreiben, aber nicht ausführen**. **Alle anderen dürfen weder lesen, schreiben noch ausführen**.

`1` gibt die Anzahl der Hardlinks an. Nach dem Gruppennamen folgt die Dateigröße. In diesem Fall handelt es sich um 8 virtuelle Blöcke und 0 reelle Blöcke. Darauf folgt der Zeitpunkt der letzten Änderung, welche am 20. April um 11:42 Uhr durchgeführt wurde. Die letzte Information ist der Dateiname, in diesem Fall `/dev/sda`.

#### 1.3.3.4 Berechtigung Zahlencodierung

Die Berechtigungen werden für jede Gruppe (Besitzer, Gruppe, Alle Anderen) als eine Zahl zusammengefasst. Somit addieren sich die einzelnen Berechtigungen zu einer Zahl zusammen. Die einzelnen Berechtigungen haben dabei folgende Werte: `r = 4`, `w = 2`, `x = 1`.

Beispielsweise könnte `rwxr-r-` als `744` dargestellt werden. Weitere Beispiele sind:

- `--x--x---` -> `110`
- `r-xr-xr-x` -> `555`

`setuid` hat den Wert 4, `setgid` den Wert 2 und `stickybit` den Wert 1. Weitere Beispiele hierzu sind:

- `rwsr--r--` -> `4744`
- `--srwS---` -> `6160`
- `r-xr-xr-t` -> `1555`
- `r-sr---wT` -> `5542`

## 1.3.4 Nutzungsverlauf

### 1.3.4.1 Unix-Zeitformat

Das Unix-Zeitformat ist auch unter den Namen Unix-Time, Epoch-Time, Posix-Time, seconds since the Epoch oder UNIX Epoch time bekannt. Es handelt sich hierbei um eine Natürliche Zahl (32 oder 64Bit), wobei die 0 dem 01. Januar 1970 um 0:00:00 Uhr UTC entspricht, was den Startzeitpunkt darstellt. Ein Zeitpunkt setzt sich aus dem Start und den seitdem vergangenen Sekunden zusammen. Schaltsekunden entstehen durch das Strecken bzw. Stauchen von Sekunden über einen langen Zeitraum.

Beispiele für Angaben im Unix-Zeitformat sind:

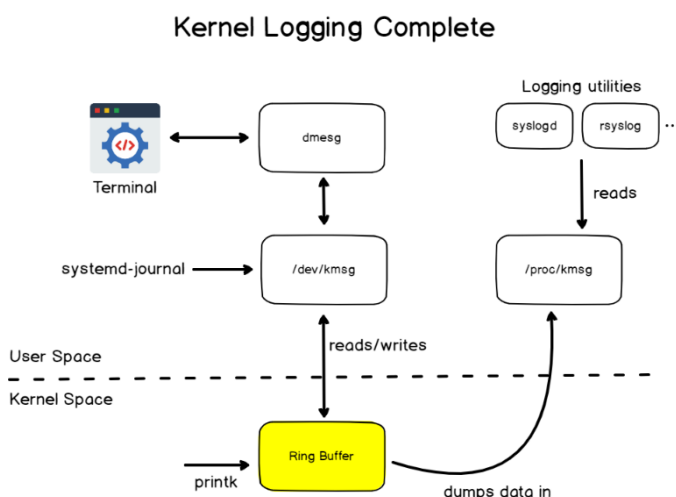
- Folie erstellt um: 1650469006
- Gründung EU: 1259625600
- Aber, Apollo 11 mit Mondlandung 20. Juni 1969 in Unix-Time nicht möglich

### 1.3.4.2 Linux-Logs

Linux-Logs protokollieren Ereignisse wie Logins, Zugriffe, auftretende Fehler oder Ähnliches. Weiterhin führen sie eine Fehlererkennung bzw. Administration durch, sodass Dateibeschädigungen oder fehlende Datenträger erkannt werden. In den Log-Dateien ist ein Eintrag in einer Zeile dargestellt:

- Timestamp
  - Unix-Time
  - ausgeschrieben in local time
  - Bootvorgang in ms nach Power on
- Akteur / Auslöser
- Eventbeschreibung
- Neuer Status

## Kernel-Logs



Der Kernel Ring Buffer speichert die Kernel-Logs, wobei `/dev/kmsg` als Interface dient. `dmesg` kann als Programm vom Endnutzer verwendet werden, um die Logs einzusehen. Eine automatische Systemauswertung erfolgt über `syslogd` (local) und `rsyslog` (remote).

## Login-Logs

Die Login-Logs befinden sich unter `/var/log/auth.log`. Die folgenden Abbildungen zeigen einige Einträge eines Login-Logs.

```
Apr 21 07:43:58 linuxmint-VirtualBox sudo: pam_unix(sudo:session): session opened for user root by (uid=0)
Apr 21 07:44:02 linuxmint-VirtualBox sudo: pam_unix(sudo:session): session closed for user root
Apr 21 08:17:01 linuxmint-VirtualBox CRON[28491]: pam_unix(cron:session): session opened for user root by (uid=0)
Apr 21 08:17:01 linuxmint-VirtualBox CRON[28491]: pam_unix(cron:session): session closed for user root
Apr 21 08:30:01 linuxmint-VirtualBox CRON[28499]: pam_unix(cron:session): session opened for user root by (uid=0)
Apr 21 08:30:01 linuxmint-VirtualBox CRON[28499]: pam_unix(cron:session): session closed for user root
Apr 21 09:05:50 linuxmint-VirtualBox su: pam_unix(su:auth): authentication failure; logname= uid=1000 euid=0 tty=pts/0
ruser=linuxmint rhost= user=root
Apr 21 09:05:51 linuxmint-VirtualBox su: FAILED SU (to root) linuxmint on pts/0
Apr 21 09:06:00 linuxmint-VirtualBox sudo: linuxmint : TTY=pts/0 ; PWD=/var/log ; USER=root ; COMMAND=/usr/bin/su
Apr 21 09:06:00 linuxmint-VirtualBox sudo: pam_unix(sudo:session): session opened for user root by (uid=0)
Apr 21 09:06:00 linuxmint-VirtualBox su: (to root) linuxmint on pts/0
Apr 21 09:06:00 linuxmint-VirtualBox su: pam_unix(su:session): session opened for user root by (uid=0)
(END)
```

sudo Befehl (Befehl mit root-Rechten)

```
Apr 21 07:43:58 linuxmint-VirtualBox sudo: pam_unix(sudo:session): session opened for user root by (uid=0)
Apr 21 07:44:02 linuxmint-VirtualBox sudo: pam_unix(sudo:session): session closed for user root
Apr 21 08:17:01 linuxmint-VirtualBox CRON[28491]: pam_unix(cron:session): session opened for user root by (uid=0)
Apr 21 08:17:01 linuxmint-VirtualBox CRON[28491]: pam_unix(cron:session): session closed for user root
Apr 21 08:30:01 linuxmint-VirtualBox CRON[28499]: pam_unix(cron:session): session opened for user root by (uid=0)
Apr 21 08:30:01 linuxmint-VirtualBox CRON[28499]: pam_unix(cron:session): session closed for user root
Apr 21 09:05:50 linuxmint-VirtualBox su: pam_unix(su:auth): authentication failure; logname= uid=1000 euid=0 tty=pts/0
ruser=linuxmint rhost= user=root
Apr 21 09:05:51 linuxmint-VirtualBox su: FAILED SU (to root) linuxmint on pts/0
Apr 21 09:06:00 linuxmint-VirtualBox sudo: linuxmint : TTY=pts/0 ; PWD=/var/log ; USER=root ; COMMAND=/usr/bin/su
Apr 21 09:06:00 linuxmint-VirtualBox sudo: pam_unix(sudo:session): session opened for user root by (uid=0)
Apr 21 09:06:00 linuxmint-VirtualBox su: (to root) linuxmint on pts/0
Apr 21 09:06:00 linuxmint-VirtualBox su: pam_unix(su:session): session opened for user root by (uid=0)
(END)
```

CRON-Job als root (Zeitgesteuerte Ausführung)

```
Apr 21 07:43:58 linuxmint-VirtualBox sudo: pam_unix(sudo:session): session opened for user root by (uid=0)
Apr 21 07:44:02 linuxmint-VirtualBox sudo: pam_unix(sudo:session): session closed for user root
Apr 21 08:17:01 linuxmint-VirtualBox CRON[28491]: pam_unix(cron:session): session opened for user root by (uid=0)
Apr 21 08:17:01 linuxmint-VirtualBox CRON[28491]: pam_unix(cron:session): session closed for user root
Apr 21 08:30:01 linuxmint-VirtualBox CRON[28499]: pam_unix(cron:session): session opened for user root by (uid=0)
Apr 21 08:30:01 linuxmint-VirtualBox CRON[28499]: pam_unix(cron:session): session closed for user root
Apr 21 09:05:50 linuxmint-VirtualBox su: pam_unix(su:auth): authentication failure; logname= uid=1000 euid=0 tty=pts/0
ruser=linuxmint rhost= user=root
Apr 21 09:05:51 linuxmint-VirtualBox su: FAILED SU (to root) linuxmint on pts/0
Apr 21 09:06:00 linuxmint-VirtualBox sudo: linuxmint : TTY=pts/0 ; PWD=/var/log ; USER=root ; COMMAND=/usr/bin/su
Apr 21 09:06:00 linuxmint-VirtualBox sudo: pam_unix(sudo:session): session opened for user root by (uid=0)
Apr 21 09:06:00 linuxmint-VirtualBox su: (to root) linuxmint on pts/0
Apr 21 09:06:00 linuxmint-VirtualBox su: pam_unix(su:session): session opened for user root by (uid=0)
(END)
```

Gescheiterter Login als root

```
Apr 21 07:43:58 linuxmint-VirtualBox sudo: pam_unix(sudo:session): session opened for user root by (uid=0)
Apr 21 07:44:02 linuxmint-VirtualBox sudo: pam_unix(sudo:session): session closed for user root
Apr 21 08:17:01 linuxmint-VirtualBox CRON[28491]: pam_unix(cron:session): session opened for user root by (uid=0)
Apr 21 08:17:01 linuxmint-VirtualBox CRON[28491]: pam_unix(cron:session): session closed for user root
Apr 21 08:30:01 linuxmint-VirtualBox CRON[28499]: pam_unix(cron:session): session opened for user root by (uid=0)
Apr 21 08:30:01 linuxmint-VirtualBox CRON[28499]: pam_unix(cron:session): session closed for user root
Apr 21 09:05:50 linuxmint-VirtualBox su: pam_unix(su:auth): authentication failure; logname= uid=1000 euid=0 tty=pts/0
ruser=linuxmint rhost= user=root
Apr 21 09:05:51 linuxmint-VirtualBox su: FAILED SU (to root) linuxmint on pts/0
Apr 21 09:06:00 linuxmint-VirtualBox sudo: linuxmint : TTY=pts/0 ; PWD=/var/log ; USER=root ; COMMAND=/usr/bin/su
Apr 21 09:06:00 linuxmint-VirtualBox sudo: pam_unix(sudo:session): session opened for user root by (uid=0)
Apr 21 09:06:00 linuxmint-VirtualBox su: (to root) linuxmint on pts/0
Apr 21 09:06:00 linuxmint-VirtualBox su: pam_unix(su:session): session opened for user root by (uid=0)
(END)
```

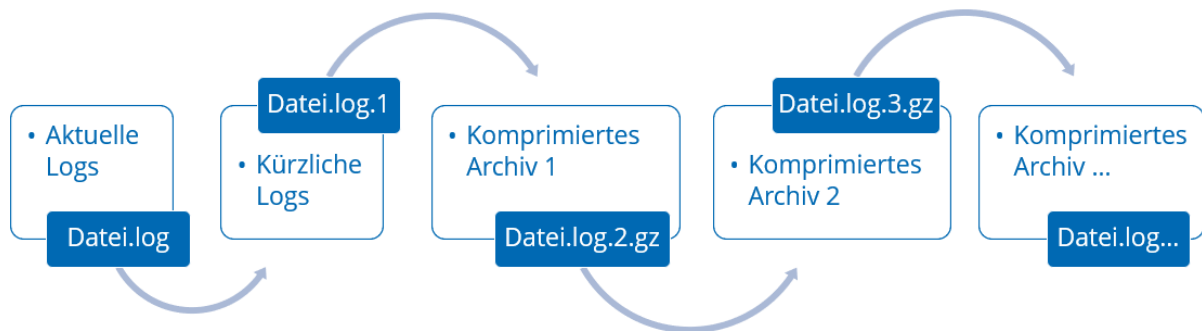
Erfolgreicher Login als root

### 1.3.4.3 Weitere Log-Speicherorte

Log-Dateien können auch an weiteren Orten gefunden werden. Im Folgenden werden weitere Fundstellen aufgelistet:

- /var/log/syslog (Ubuntu/Debian) bzw. /var/log/messages (Red Hat)
  - Allgemeine System Logs
- /var/log/auth.log (Ubuntu/Debian) bzw. /var/log/secure (Red Hat)
  - Authentifikation (Login)
- /var/log/maillog bzw. /var/log/mail.log
  - E-Mail
- /var/log/kern
  - Kernel-Logs
- /var/log/dmesg
  - Kernel-Logs über Geräte
- /var/log/faillog
  - Gescheiterte Logins Zusammenfassung (Befehl: faillog)
- /var/log/cron
  - Logs von Cron-Jobs
- /var/log/daemon.log
  - Logs von Hintergrundprozessen
- /var/log/httpd/
  - Webserver-Logs
- /var/log/mysql.log
  - SQL-Server Logs
- /var/log/xferlog
  - Logs von FTP-Übertragungen

#### 1.3.4.4 Log-Archiv



#### 1.3.4.5 Shell-Historie

Die Shell-Historie speichert ausgeführte Befehle der Shell, jedoch werden diese nicht mit einem Zeitstempel abgelegt, sodass eine Korrelation mit anderen Logs notwendig ist. Die Shell-Historie wird unter `~/.bash_history` abgespeichert. Diese Datei wird beim Logout oder bei Schließen des Shell-Fensters beschrieben. Vorher lassen sich nur Informationen aus dem RAM auslesen. Dies kann über den Befehl „history“ erfolgen, wobei jede Shell-Instanz eine eigene History hat. Diese History wird während des Nutzens der Shell verwendet, wenn man über die Pfeiltasten ältere Befehle anzeigen lässt.

Die Shell-Historie kann auch als Indicator of Compromise dienen. In diesem Fall ist beispielsweise die History durch cron geleert, die Datei `~/.bash_history` wird durch einen Softlink, z.B. auf `/dev/null`, ersetzt oder das Änderungsdatum der Datei `~/.bash_history` korreliert nicht mit dem logout. Dies lässt auf eine manuelle Manipulation schließen, wobei möglicherweise der Befehl gelöscht wurde.

#### 1.3.4.6 Dateizugriff

Der Dateizugriff ist im Filesystem gespeichert, wobei einige Filesysteme diese Metadaten nicht unterstützen. Den letzten Schreibzugriff auf eine Datei erhält man mit den Befehlen „`fs -l <Datei>`“ oder „`date -r <Datei>`“. Ausführlichere Informationen zu einer Datei erhält man über den Befehl „`stat <Datei>`“. Dabei sollte aber immer beachtet werden, dass der root-Nutzer alles ändern kann.

```

linuxmint@linuxmint-VirtualBox:~/disks$ ls -l
insgesamt 393268
-rw-rw-r-- 1 linuxmint linuxmint 134217728 Apr 20 15:05 disk1
-rw-rw-r-- 1 linuxmint linuxmint 134217728 Apr 20 15:05 disk2
-rw-rw-r-- 1 linuxmint linuxmint 134217728 Apr 20 15:05 disk3
-rw-rw-r-- 1 linuxmint linuxmint 38349 Apr 20 11:39 manpage.txt
-rw-rw-r-- 1 linuxmint linuxmint 2 Apr 20 17:12 test
linuxmint@linuxmint-VirtualBox:~/disks$ date -r disk1
Mi 20 Apr 2022 15:05:15 CEST
linuxmint@linuxmint-VirtualBox:~/disks$ stat disk1
  Datei: disk1
  Größe: 134217728      Blöcke: 262152      EA Block: 4096   Normale Datei
Gerät: fd00h/64768d    Inode: 1059378     Verknüpfungen: 1
Zugriff: (0664/-rw-rw-r-- ) Uid: ( 1000/linuxmint)  Gid: ( 1000/linuxmint)
Zugriff: 2022-04-20 15:05:09.376448864 +0200
Modifiziert: 2022-04-20 15:05:15.360480843 +0200
Geändert: 2022-04-20 15:05:15.360480843 +0200
Geburt: -

```

#### 1.3.4.7 Systemschnappschüsse

Für Systemschnappschüsse kann timeshift als Softwarelösung eingesetzt werden. Dieses Tool ist vergleichbar mit der Windows Systemwiederherstellung oder der Apple Time Machine. Für die Systemschnappschüsse wird rsync genutzt, was für die Synchronisierung von Dateien zuständig ist. Bei timeshift werden hier nur die Änderungen zur vorherigen Version gespeichert. Es wurde für die Speicherstände von Einstellungen und Systemdateien entworfen. Eine Automatisierung für die tägliche, wöchentliche oder monatliche Durchführung von Schnappschüssen ist möglich.

#### 1.3.5 Zusammenfassung

Sie kennen nun Speicherorte und den Aufbau der Netzwerk-Konfigurations-Dateien unter Linux. Sie können außerdem die WLAN-Konfiguration unter verschiedenen Distributionen auswerten.

Heute haben Sie einen Überblick über Dateisysteme und Partitions-möglichkeiten erhalten. Sie kennen darüber hinaus die notwendigen Programme zum Partitionieren und Formatieren. Sie wissen über die RAID-Versionen 0, 1, 5 und 10 bescheid.

Sie wissen nun, wie Nutzer unter Linux gespeichert werden und wie deren Rechtevergabe funktioniert. Die Dateifreigabe können Sie interpretieren.

Auf einem System können Sie Spuren des Nutzungsverlaufs auswerten. Den Aufbau von Logs unter Linux haben Sie heute gelernt und können gegebenenfalls Zeitangaben in Unix-Time in einem Zeitpunkt umrechnen.

## 1.4 Linux Anwendungen im Terminal I

### 1.4.1 Linux Shell

Die Bash ist die meistgenutzte Shell. Die Implementierung der Bash ist in der IEEE POSIX Spezifikation (IEEE Standard 1003.1) beschrieben. Die Bash interpretiert verschiedenste Befehle. Dazu gehören Befehle für den Programmstart, die Ein- und Ausgabeverwaltung, die Umleitung von Ein- und Ausgaben sowie welche für die Skriptumgebungen. Weiterhin definiert die Bash die Befehlssyntax, sprich den Aufbau der Befehle. Eine Benutzerinteraktion ist möglich. Beispielsweise kann der Nutzer selbst Befehle abbrechen oder verschiedenste Tastenkürzel nutzen. Für den Benutzerkomfort ist die Darstellung von Links möglich und eine Autovervollständigung des Befehls implementiert.

Neben der Bash gibt es noch weitere Shells. sh (Shell) ist die minimale Shell. csh (C Shell) ist eine Simple Shell, wird mittlerweile jedoch obsolet, da es schon einen Nachfolger gibt. Die ksh (Korn Shell) bietet eine volle Kompatibilität mit der Bash und setzt ihren Fokus auf die Performance und die Programmierbarkeit. Weiterhin gibt es die zsh (Z Shell), welche der Bash ähnlich ist, aber noch andere Zusatzfunktionen implementiert hat. fish legt den Fokus auf die Benutzerinteraktion, wohingegen dash den Fokus auf die Geschwindigkeit und den Ressourcenbedarf legt. Die Shell xonsh ist eine Kooperation mit Python. Zuletzt gibt es noch nushell, welche vor allem für die Formatierung von Ausgaben in Tabellenform genutzt wird.

### 1.4.2 Navigation

Die Navigation innerhalb des Systems erfolgt mit verschiedensten Befehlen. Diese werden im Folgenden näher erläutert.

#### 1.4.2.1 ls

Der Befehl ls ist die Kurzform für list und wird für das Auflisten von Dateien angewendet. Er kann mit Parametern erweitert werden, um die Ausgabe zu verfeinern. Der Parameter -l (list) bietet eine zeilenweise Auflistung der Dateien. -i (inode) zeigt die Indexierung der Dateien an. -s (size) gibt zusätzlich die Dateigröße aus. -a (all) gibt alle Dateien aus, sprich auch Dateien, welche versteckt sind. Der Parameter -h steht für human readable und gibt die Größe in Si-Einheiten an.



```
fische11@DESKTOP-5MHG39V: ~  
fische11@DESKTOP-5MHG39V:~$ ls -liah  
total 1.3M  
664 4.0K drwxr-xr-x 8 fische11 fische11 4.0K May 4 15:13 .  
16386 4.0K drwxr-xr-x 3 root root 4.0K Mar 2 17:33 ..  
77 8.0K -rw----- 1 fische11 fische11 6.1K May 4 15:24 .bash_history  
40060 4.0K -rw-r--r-- 1 fische11 fische11 220 Mar 2 17:33 .bash_logout  
40063 4.0K -rw-r--r-- 1 fische11 fische11 3.7K Mar 2 17:33 .bashrc  
34483 4.0K -rw-r--r-- 1 fische11 fische11 21 Apr 27 21:22 .gitconfig  
40059 4.0K drwxr-xr-x 2 fische11 fische11 4.0K Mar 2 17:33 .landscape  
34473 4.0K drwxr-xr-x 3 fische11 fische11 4.0K Apr 27 21:20 .local  
40068 0 -rw-r--r-- 1 fische11 fische11 0 May 4 13:02 .motd_shown  
1345 4.0K -rw-r--r-- 1 fische11 fische11 807 Mar 2 17:33 .profile  
238 4.0K drwx----- 2 fische11 fische11 4.0K Apr 22 18:21 .ssh  
40069 0 -rw-r--r-- 1 fische11 fische11 0 Mar 2 17:34 .sudo_as_admin_successful  
649 4.0K drwxr-xr-x 3 fische11 fische11 4.0K Apr 27 23:44 Uebung_Hash  
34540 588K -rwxr-xr-x 1 fische11 fische11 585K Apr 28 10:18 Uebung_Hash.tar.gz  
635 600K -rwxr-xr-x 1 fische11 fische11 599K Apr 27 20:21 Uebung_Hash.zip  
34516 4.0K -rwxr--r-- 1 fische11 fische11 1.1K Apr 28 09:46 basic_info.sh  
403 4.0K -rw-r--r-- 1 fische11 fische11 6 May 4 15:13 bla  
34528 4.0K -rw-r--r-- 1 fische11 fische11 111 Apr 28 09:29 error.log  
34502 4.0K drwxr-xr-x 2 fische11 fische11 4.0K Apr 28 09:24 main_infos  
12967 4.0K drwxr-xr-x 3 fische11 fische11 4.0K Apr 27 21:25 test  
34544 4.0K -rwxr--r-- 1 fische11 fische11 731 Apr 28 11:16 unknown.sh  
fische11@DESKTOP-5MHG39V:~$
```

#### 1.4.2.2 cd


Der Befehl `cd` ist die Kurzform für `change directory` und wird für den Verzeichniswechsel genutzt. Man kann über den absoluten oder den relativen Pfad in ein Verzeichnis wechseln. Mit dem Befehl „`cd ~`“ oder nur „`cd`“ wechselt man zurück in das eigene Home-Verzeichnis.

```
fische11@DESKTOP-5MHG39V: ~  
fische11@DESKTOP-5MHG39V:~$ cd test/  
fische11@DESKTOP-5MHG39V:~/test$ cd /var/log  
fische11@DESKTOP-5MHG39V:/var/log$ cd ../../etc  
fische11@DESKTOP-5MHG39V:/etc$ cd ~  
fische11@DESKTOP-5MHG39V:~$ cd .ssh/  
fische11@DESKTOP-5MHG39V:~/.ssh$ cd  
fische11@DESKTOP-5MHG39V:~$
```



### 1.4.2.3 pwd

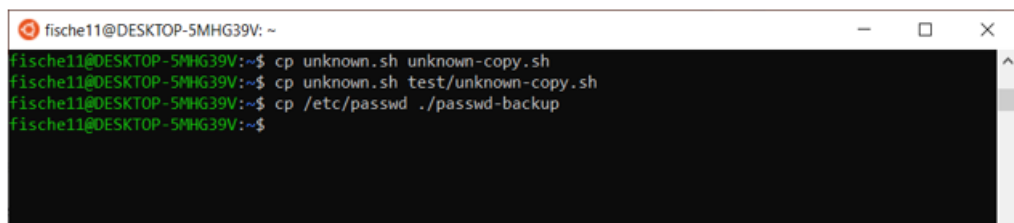
Der Befehl `pwd` ist die Kurzform für `print working directory`. Er wird für die Anzeige des aktuellen Verzeichnispfades angewendet.



```
fische11@DESKTOP-5MHG39V: ~/ssh
fische11@DESKTOP-5MHG39V:~$ pwd
/home/fische11
fische11@DESKTOP-5MHG39V:~$ cd /etc/
fische11@DESKTOP-5MHG39V:/etc$ pwd
/etc
fische11@DESKTOP-5MHG39V:/etc$ cd ~/.ssh
fische11@DESKTOP-5MHG39V:~/.ssh$ pwd
/home/fische11/.ssh
fische11@DESKTOP-5MHG39V:~/.ssh$
```

### 1.4.2.4 cp

Der Befehl `cp` ist die Kurzform für `copy` und wird für das Kopieren von Dateien genutzt. Der Aufbau des Befehls ist `cp <Original> <Kopie>`. Jedoch muss man aufpassen, da hier Daten ohne Warnung überschrieben werden. Es können zusätzliche Parameter angegeben werden. Mit `-l` wird ein `hard-link` erstellt, anstatt die Originaldatei zu kopieren. Mit `-s` wird hingegen ein `soft-link` erstellt, statt die Datei zu kopieren.



```
fische11@DESKTOP-5MHG39V: ~
fische11@DESKTOP-5MHG39V:~$ cp unknown.sh unknown-copy.sh
fische11@DESKTOP-5MHG39V:~$ cp unknown.sh test/unknown-copy.sh
fische11@DESKTOP-5MHG39V:~$ cp /etc/passwd ./passwd-backup
fische11@DESKTOP-5MHG39V:~$
```

### 1.4.2.5 mv

Der Befehl `mv` ist die Kurzform für `move` und wird für das Verschieben und Umbenennen von Dateien eingesetzt. `mv` überschreibt die Zieldatei. Mit dem Parameter `-b` wird ein Backup der Zieldatei erstellt, falls diese bereits existiert.

```

fische11@DESKTOP-5MHG39V: ~
fische11@DESKTOP-5MHG39V:~$ mv unknown
unknown-copy.sh unknown.sh
fische11@DESKTOP-5MHG39V:~$ mv unknown-copy.sh unknown-kopie.sh
fische11@DESKTOP-5MHG39V:~$ mv -b unknown-kopie.sh unknown.sh
fische11@DESKTOP-5MHG39V:~$ ls unknown*
unknown.sh unknown.sh~
fische11@DESKTOP-5MHG39V:~$

```

Backup von unknown.sh

Ursprünglich unknown-kopie.sh

#### 1.4.2.6 touch

Der Befehl touch wird für das „Anfassen“ der Datei genutzt. Hierbei wird die letzte Zugriffszeit auf die aktuelle Uhrzeit gesetzt. Falls die Datei nicht vorhanden ist, wird eine leere Datei erstellt. touch kann mit weiteren Parametern ausgeführt werden. Mit dem Parameter -t kann die Zugriffszeit selbst festgelegt werden. Mit -c wird keine leere Datei erstellt.

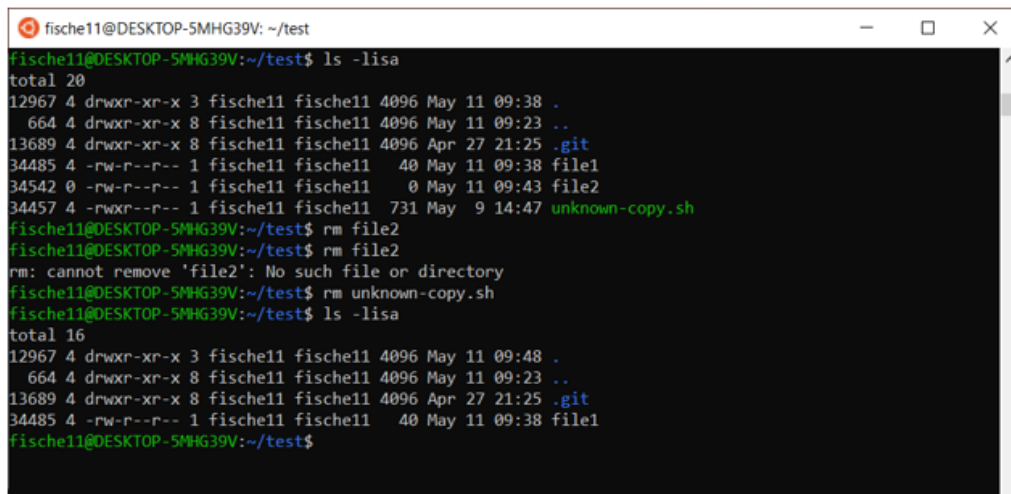
```

fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~$ cd test/
fische11@DESKTOP-5MHG39V:~/test$ ls
file1 unknown-copy.sh
fische11@DESKTOP-5MHG39V:~/test$ touch file2
fische11@DESKTOP-5MHG39V:~/test$ ls
file1 file2 unknown-copy.sh
fische11@DESKTOP-5MHG39V:~/test$ ls -lisa
total 20
12967 4 drwxr-xr-x 3 fische11 fische11 4096 May 11 09:38 .
 664 4 drwxr-xr-x 8 fische11 fische11 4096 May 11 09:23 ..
13689 4 drwxr-xr-x 8 fische11 fische11 4096 Apr 27 21:25 .git
34485 4 -rw-r--r-- 1 fische11 fische11 40 Apr 27 21:25 file1
34542 0 -rw-r--r-- 1 fische11 fische11 0 May 11 09:38 file2
34457 4 -rwxr--r-- 1 fische11 fische11 731 May 9 14:47 unknown-copy.sh
fische11@DESKTOP-5MHG39V:~/test$ touch file1
fische11@DESKTOP-5MHG39V:~/test$ ls -lisa
total 20
12967 4 drwxr-xr-x 3 fische11 fische11 4096 May 11 09:38 .
 664 4 drwxr-xr-x 8 fische11 fische11 4096 May 11 09:23 ..
13689 4 drwxr-xr-x 8 fische11 fische11 4096 Apr 27 21:25 .git
34485 4 -rw-r--r-- 1 fische11 fische11 40 May 11 09:38 file1
34542 0 -rw-r--r-- 1 fische11 fische11 0 May 11 09:38 file2
34457 4 -rwxr--r-- 1 fische11 fische11 731 May 9 14:47 unknown-copy.sh
fische11@DESKTOP-5MHG39V:~/test$ touch -c file3
fische11@DESKTOP-5MHG39V:~/test$ ls -lisa
total 20
12967 4 drwxr-xr-x 3 fische11 fische11 4096 May 11 09:38 .
 664 4 drwxr-xr-x 8 fische11 fische11 4096 May 11 09:23 ..
13689 4 drwxr-xr-x 8 fische11 fische11 4096 Apr 27 21:25 .git
34485 4 -rw-r--r-- 1 fische11 fische11 40 May 11 09:38 file1
34542 0 -rw-r--r-- 1 fische11 fische11 0 May 11 09:38 file2
34457 4 -rwxr--r-- 1 fische11 fische11 731 May 9 14:47 unknown-copy.sh
fische11@DESKTOP-5MHG39V:~/test$ touch -c file2
fische11@DESKTOP-5MHG39V:~/test$ ls -lisa
total 20
12967 4 drwxr-xr-x 3 fische11 fische11 4096 May 11 09:38 .
 664 4 drwxr-xr-x 8 fische11 fische11 4096 May 11 09:23 ..
13689 4 drwxr-xr-x 8 fische11 fische11 4096 Apr 27 21:25 .git
34485 4 -rw-r--r-- 1 fische11 fische11 40 May 11 09:38 file1
34542 0 -rw-r--r-- 1 fische11 fische11 0 May 11 09:43 file2
34457 4 -rwxr--r-- 1 fische11 fische11 731 May 9 14:47 unknown-copy.sh
fische11@DESKTOP-5MHG39V:~/test$

```

### 1.4.2.7 rm

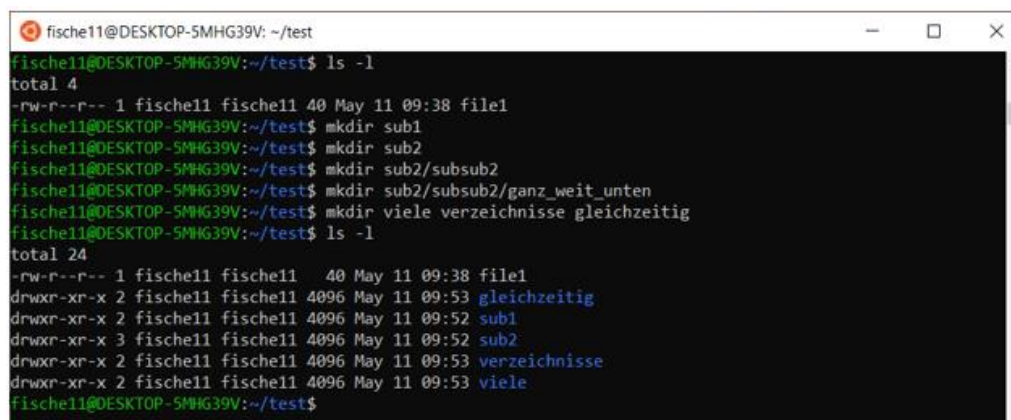
Der Befehl `rm` ist die Kurzform für `remove` und wird für das Löschen von Dateien genutzt. Der Parameter `-r` steht für `recursive`, sodass auch Unterordner mit gelöscht werden. `-f` steht für `force` und erzwingt das Löschen. Vorsicht ist bei folgenden Befehlen geboten: „`rm -rf /`“ und „`sudo rm -rf /`“.



```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ ls -lisa
total 20
12967 4 drwxr-xr-x 3 fische11 fische11 4096 May 11 09:38 .
 664 4 drwxr-xr-x 8 fische11 fische11 4096 May 11 09:23 ..
13689 4 drwxr-xr-x 8 fische11 fische11 4096 Apr 27 21:25 .git
34485 4 -rw-r--r-- 1 fische11 fische11 40 May 11 09:38 file1
34542 0 -rw-r--r-- 1 fische11 fische11 0 May 11 09:43 file2
34457 4 -rwxr--r-- 1 fische11 fische11 731 May 9 14:47 unknown-copy.sh
fische11@DESKTOP-5MHG39V:~/test$ rm file2
fische11@DESKTOP-5MHG39V:~/test$ rm file2
rm: cannot remove 'file2': No such file or directory
fische11@DESKTOP-5MHG39V:~/test$ rm unknown-copy.sh
fische11@DESKTOP-5MHG39V:~/test$ ls -lisa
total 16
12967 4 drwxr-xr-x 3 fische11 fische11 4096 May 11 09:48 .
 664 4 drwxr-xr-x 8 fische11 fische11 4096 May 11 09:23 ..
13689 4 drwxr-xr-x 8 fische11 fische11 4096 Apr 27 21:25 .git
34485 4 -rw-r--r-- 1 fische11 fische11 40 May 11 09:38 file1
fische11@DESKTOP-5MHG39V:~/test$
```

### 1.4.2.8 mkdir

Der Befehl `mkdir` ist die Kurzform für `make directory`. Mit diesem Befehl werden Ordner erstellt.



```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ ls -l
total 4
-rw-r--r-- 1 fische11 fische11 40 May 11 09:38 file1
fische11@DESKTOP-5MHG39V:~/test$ mkdir sub1
fische11@DESKTOP-5MHG39V:~/test$ mkdir sub2
fische11@DESKTOP-5MHG39V:~/test$ mkdir sub2/subsub2
fische11@DESKTOP-5MHG39V:~/test$ mkdir sub2/subsub2/ganz_weit_unten
fische11@DESKTOP-5MHG39V:~/test$ mkdir viele verzeichnisse gleichzeitig
fische11@DESKTOP-5MHG39V:~/test$ ls -l
total 24
-rw-r--r-- 1 fische11 fische11 40 May 11 09:38 file1
drwxr-xr-x 2 fische11 fische11 4096 May 11 09:53 gleichzeitig
drwxr-xr-x 2 fische11 fische11 4096 May 11 09:52 sub1
drwxr-xr-x 3 fische11 fische11 4096 May 11 09:52 sub2
drwxr-xr-x 2 fische11 fische11 4096 May 11 09:53 verzeichnisse
drwxr-xr-x 2 fische11 fische11 4096 May 11 09:53 viele
fische11@DESKTOP-5MHG39V:~/test$
```

### 1.4.2.9 rmdir

Der Befehl `rmdir` ist die Kurzform für `remove directory` und ist für das Löschen von Ordnern zuständig. Mit dem Parameter `-p` werden alle Ordner bis zu den Unterordnern gelöscht.

```

fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ ls -l
total 24
-rw-r--r-- 1 fische11 fische11 40 May 11 09:38 file1
drwxr-xr-x 2 fische11 fische11 4096 May 11 09:53 gleichzeitig
drwxr-xr-x 2 fische11 fische11 4096 May 11 09:59 sub1
drwxr-xr-x 3 fische11 fische11 4096 May 11 09:59 sub2
drwxr-xr-x 2 fische11 fische11 4096 May 11 09:53 verzeichnisse
drwxr-xr-x 2 fische11 fische11 4096 May 11 09:53 viele
fische11@DESKTOP-5MHG39V:~/test$ rmdir sub1
fische11@DESKTOP-5MHG39V:~/test$ rmdir sub2
rmdir: failed to remove 'sub2': Directory not empty
fische11@DESKTOP-5MHG39V:~/test$ rmdir sub2/subsub2/ganz_weit_unten
fische11@DESKTOP-5MHG39V:~/test$ rmdir -p sub2/subsub2
fische11@DESKTOP-5MHG39V:~/test$ ls -l
total 16
-rw-r--r-- 1 fische11 fische11 40 May 11 09:38 file1
drwxr-xr-x 2 fische11 fische11 4096 May 11 09:53 gleichzeitig
drwxr-xr-x 2 fische11 fische11 4096 May 11 09:53 verzeichnisse
drwxr-xr-x 2 fische11 fische11 4096 May 11 09:53 viele
fische11@DESKTOP-5MHG39V:~/test$ rmdir viele verzeichnisse gleichzeitig
fische11@DESKTOP-5MHG39V:~/test$ ls -l
total 4
-rw-r--r-- 1 fische11 fische11 40 May 11 09:38 file1
fische11@DESKTOP-5MHG39V:~/test$

```

#### 1.4.2.10 ln

Der Befehl ln ist die Kurzform für link und ist für das Erstellen von Links zuständig. Ein Link entspricht einer Verknüpfung unter Windows. Mit dem Befehl wird die Zieldatei (angegebener Linkname) überschrieben. Der Befehl ist wie folgt aufgebaut: ln <datei> <linkname>. Standardmäßig wird ein Hardlink erstellt, jedoch kann mit dem Parameter -s auch ein Softlink erstellt werden.

```

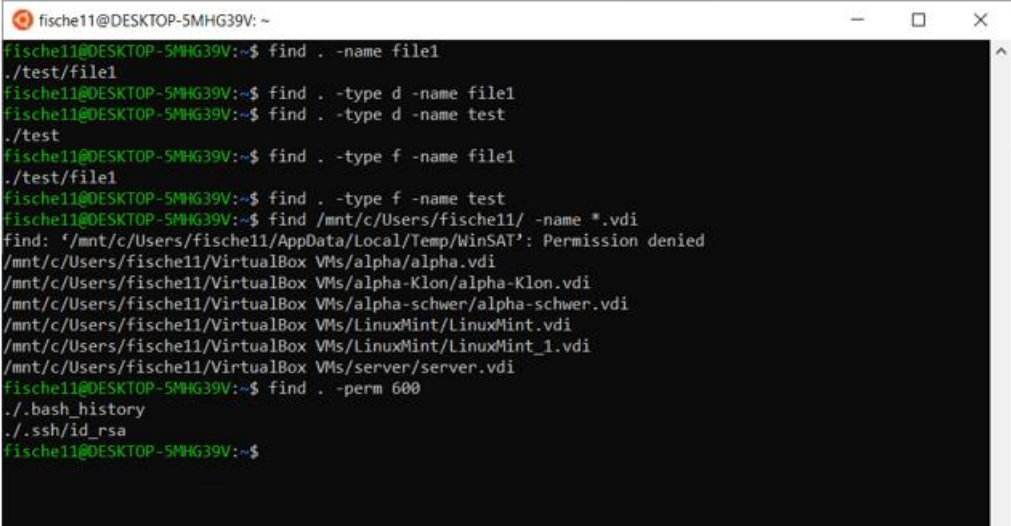
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ ls -lisa
total 16
12967 4 drwxr-xr-x 3 fische11 fische11 4096 May 11 10:19 .
 664 4 drwxr-xr-x 8 fische11 fische11 4096 May 11 09:23 ..
13689 4 drwxr-xr-x 8 fische11 fische11 4096 Apr 27 21:25 .git
34485 4 -rw-r--r-- 1 fische11 fische11 40 May 11 09:38 file1
fische11@DESKTOP-5MHG39V:~/test$ ln file1 hardlink
fische11@DESKTOP-5MHG39V:~/test$ ln -s file1 softlink
fische11@DESKTOP-5MHG39V:~/test$ ls -lisa
total 20
12967 4 drwxr-xr-x 3 fische11 fische11 4096 May 11 10:20 .
 664 4 drwxr-xr-x 8 fische11 fische11 4096 May 11 09:23 ..
13689 4 drwxr-xr-x 8 fische11 fische11 4096 Apr 27 21:25 .git
34485 4 -rw-r--r-- 2 fische11 fische11 40 May 11 09:38 file1
34485 4 -rw-r--r-- 2 fische11 fische11 40 May 11 09:38 hardlink
34457 0 lrwxrwxrwx 1 fische11 fische11 5 May 11 10:20 softlink -> file1
fische11@DESKTOP-5MHG39V:~/test$

```

Hardlinkanzahl

### 1.4.2.11 find

Der Befehl find ist für das Finden von Dateien zuständig. Der Befehlsaufbau ist wie folgt: find <start-ordner> <optionen>. Als Optionen kann -name genutzt werden, um den Dateinamen der gesuchten Datei anzugeben. Mit -type kann man außerdem den Dateitypen angeben, wobei d für directory steht und f für file. Weiterhin können mit -perm noch Dateirechte und mit -links die Anzahl der Hardlinks angegeben werden.



```
fische11@DESKTOP-5MHG39V: ~  
fische11@DESKTOP-5MHG39V:~$ find . -name file1  
./test/file1  
fische11@DESKTOP-5MHG39V:~$ find . -type d -name file1  
fische11@DESKTOP-5MHG39V:~$ find . -type d -name test  
./test  
fische11@DESKTOP-5MHG39V:~$ find . -type f -name file1  
./test/file1  
fische11@DESKTOP-5MHG39V:~$ find . -type f -name test  
fische11@DESKTOP-5MHG39V:~$ find /mnt/c/Users/fische11/ -name *.vdi  
find: '/mnt/c/Users/fische11/AppData/Local/Temp/WinSAT': Permission denied  
/mnt/c/Users/fische11/VirtualBox VMs/alpha/alpha.vdi  
/mnt/c/Users/fische11/VirtualBox VMs/alpha-Klon/alpha-Klon.vdi  
/mnt/c/Users/fische11/VirtualBox VMs/alpha-schwer/alpha-schwer.vdi  
/mnt/c/Users/fische11/VirtualBox VMs/LinuxMint/LinuxMint.vdi  
/mnt/c/Users/fische11/VirtualBox VMs/LinuxMint/LinuxMint_1.vdi  
/mnt/c/Users/fische11/VirtualBox VMs/server/server.vdi  
fische11@DESKTOP-5MHG39V:~$ find . -perm 600  
./.bash_history  
./.ssh/id_rsa  
fische11@DESKTOP-5MHG39V:~$
```

### 1.4.3 Dateiinhalt

Die folgenden Befehle können genutzt werden, um den Inhalt von Dateien zu betrachten.

#### 1.4.3.1 cat

Der Befehl „cat“ ist die Kurzform von concatenate und ist für die Ausgabe von Dateien zuständig. Der Befehl „tac“ gibt den Inhalt der Datei zeilenweise in umgekehrter Reihenfolge aus.

```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ cat file1
Das ist die Datei file1.
Nur ein wenig Text.
Eine weitere Zeile.
fische11@DESKTOP-5MHG39V:~/test$ cat file2
Das ist die Datei file2.
Hier steht auch etwas Text.

Nach der Leere kommt das:
Ende
fische11@DESKTOP-5MHG39V:~/test$ cat file1 file2
Das ist die Datei file1.
Nur ein wenig Text.
Eine weitere Zeile.
Das ist die Datei file2.
Hier steht auch etwas Text.

Nach der Leere kommt das:
Ende
fische11@DESKTOP-5MHG39V:~/test$ tac file1
Eine weitere Zeile.
Nur ein wenig Text.
Das ist die Datei file1.
fische11@DESKTOP-5MHG39V:~/test$ tac file2
Ende
Nach der Leere kommt das:

Hier steht auch etwas Text.
Das ist die Datei file2.
fische11@DESKTOP-5MHG39V:~/test$ tac file1 file2
Eine weitere Zeile.
Nur ein wenig Text.
Das ist die Datei file1.
Ende
Nach der Leere kommt das:

Hier steht auch etwas Text.
```

### 1.4.3.2 hexdump

Der Befehl „hexdump“ gibt die Bytes einer Datei aus. Mit dem Parameter -C werden die Bytes als ASCII Zeichen interpretiert, wobei das Zeichen ausgegeben wird, wenn es gültig ist. Ansonsten erscheint ein „.“.



```

fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ hexdump file1
00000000 6144 2073 7369 2074 6964 2065 6144 6574
00000010 2069 6966 656c 2e31 4e8a 7275 6520 6e69
00000020 7720 6e65 6769 5420 7865 2e74 450a 6e69
00000030 2065 6577 7469 7265 2065 655a 6c69 2e65
00000040 000a
00000041
fische11@DESKTOP-5MHG39V:~/test$ hexdump -C file1
00000000 44 61 73 20 69 73 74 20 64 69 65 20 44 61 74 65 |Das ist die Date|
00000010 09 20 66 69 6c 65 31 2e 0a 4e 75 72 20 65 69 6e |i file1. Nur ein|
00000020 20 77 65 6e 69 67 20 54 65 78 74 2e 0a 45 6a 6a |wenig Text. Ein|
00000030 65 20 77 65 69 74 65 72 65 20 5a 65 69 6c 65 2e |e weitere Zeile.|
00000040 0a
00000041
fische11@DESKTOP-5MHG39V:~/test$ hexdump -C /bin/cp
00000000 7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00 |.ELF.....|
00000010 03 00 3e 00 01 00 00 00 d0 5a 00 00 00 00 00 00 |.|>.....Z.....|
00000020 40 00 00 00 00 00 00 00 f8 51 02 00 00 00 00 00 |@.....Q.....|
00000030 00 00 00 40 00 38 00 0d 00 40 00 1e 00 1d 00 00 |...@.B...@.....|
00000040 06 00 00 04 00 00 00 40 00 00 00 00 00 00 00 00 |@.....@.....|
00000050 40 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00 |@.....@.....|
00000060 d8 02 00 00 00 00 00 d8 02 00 00 00 00 00 00 00 |@.....@.....|
00000070 08 00 00 00 00 00 00 03 00 00 00 04 00 00 00 00 |@.....@.....|
00000080 18 03 00 00 00 00 00 18 03 00 00 00 00 00 00 00 |@.....@.....|
00000090 18 03 00 00 00 00 00 1c 00 00 00 00 00 00 00 00 |@.....@.....|
000000a0 1c 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00 |@.....@.....|
000000b0 01 00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 |@.....@.....|
000000c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |@.....@.....|
000000d0 88 34 00 00 00 00 00 88 34 00 00 00 00 00 00 00 |.4.....4.....|
000000e0 00 10 00 00 00 00 00 01 00 00 00 05 00 00 00 00 |@.....@.....|
000000f0 00 40 00 00 00 00 00 00 40 00 00 00 00 00 00 00 |@.....@.....|
00000100 00 40 00 00 00 00 00 01 74 01 00 00 00 00 00 00 |@.....@.....|
00000110 01 74 01 00 00 00 00 00 10 00 00 00 00 00 00 00 |@.....@.....|
00000120 01 00 00 04 00 00 00 00 c0 01 00 00 00 00 00 00 |@.....@.....|
00000130 00 c0 01 00 00 00 00 00 00 c0 01 00 00 00 00 00 00 |@.....@.....|
00000140 50 75 00 00 00 00 00 50 75 00 00 00 00 00 00 00 |Pu.....Pu.....|
00000150 00 10 00 00 00 00 00 00 01 00 00 06 00 00 00 00 |@.....@.....|
00000160 d0 43 02 00 00 00 00 d0 53 02 00 00 00 00 00 00 |.C.....S.....|
00000170 d0 53 02 00 00 00 00 d0 0c 00 00 00 00 00 00 00 |.S.....|
00000180 48 13 00 00 00 00 00 00 10 00 00 00 00 00 00 00 |H.....H.....|
00000190 02 00 00 06 00 00 00 f8 48 02 00 00 00 00 00 00 |X.....X.....|
000001a0 f8 53 02 00 00 00 00 f8 53 02 00 00 00 00 00 00 |X.....X.....|
000001b0 20 02 00 00 00 00 00 20 02 00 00 00 00 00 00 00 |@.....@.....|
000001c0 08 00 00 00 00 00 00 04 00 00 00 04 00 00 00 00 |@.....@.....|
000001d0 38 03 00 00 00 00 00 38 03 00 00 00 00 00 00 00 |8.....8.....|
000001e0 38 03 00 00 00 00 00 20 00 00 00 00 00 00 00 00 |8.....8.....|
000001f0 20 00 00 00 00 00 00 08 00 00 00 00 00 00 00 00 |@.....@.....|
00000200 04 00 00 04 00 00 00 58 03 00 00 00 00 00 00 00 |X.....X.....|
00000210 58 03 00 00 00 00 00 58 03 00 00 00 00 00 00 00 |X.....X.....|
00000220 44 00 00 00 00 00 00 44 00 00 00 00 00 00 00 00 |D.....D.....|

```

### 1.4.3.3 more

Der Befehl „more“ unterbricht die Ausgabe, wenn der Bildschirm voll ist. Der Befehlsaufbau ist „more <Datei>“. Mit Enter kann man eine Zeile weitersehen und mit der Leertaste wird eine ganze neue Seite der Ausgabe angezeigt. Häufig erfolgt die Eingabe von vorherigen Programmen, wobei diese mit more über eine Pipe (|) verbunden werden.

```

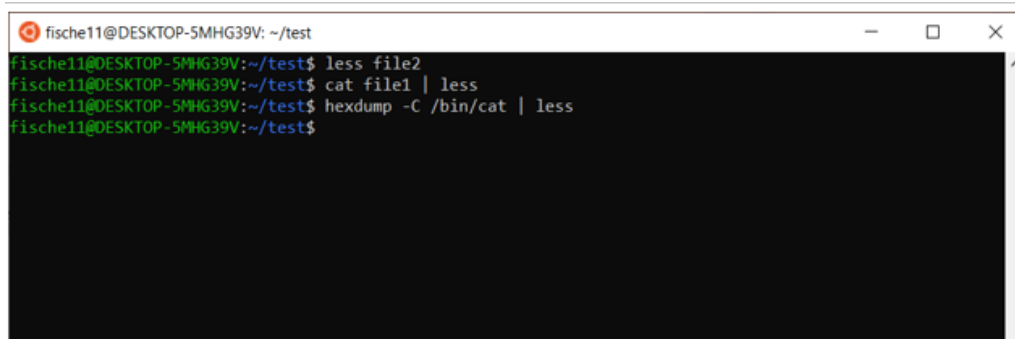
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ cat file1 | more
Das ist die Datei file1.
Nur ein wenig Text.
Eine weitere Zeile.
fische11@DESKTOP-5MHG39V:~/test$ more file2
Das ist die Datei file2.
Hier steht auch etwas Text.

Nach der Leere kommt das:
Ende
fische11@DESKTOP-5MHG39V:~/test$ hexdump -C /bin/cat | more
00000000 7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00 |.ELF.....|
00000010 03 00 3e 00 01 00 00 00 f0 31 00 00 00 00 00 00 |.|>.....1.....|
00000020 40 00 00 00 00 00 00 00 18 a2 00 00 00 00 00 00 |@.....@.....|
00000030 00 00 00 40 00 38 00 0d 00 40 00 1e 00 1d 00 00 |...@.B...@.....|
00000040 06 00 00 04 00 00 00 40 00 00 00 00 00 00 00 00 |@.....@.....|
00000050 40 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00 |@.....@.....|
00000060 d8 02 00 00 00 00 00 d8 02 00 00 00 00 00 00 00 |@.....@.....|
00000070 08 00 00 00 00 00 00 03 00 00 00 04 00 00 00 00 |@.....@.....|
00000080 18 03 00 00 00 00 00 18 03 00 00 00 00 00 00 00 |@.....@.....|
00000090 18 03 00 00 00 00 00 1c 00 00 00 00 00 00 00 00 |@.....@.....|
000000a0 1c 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00 |@.....@.....|
000000b0 01 00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 |@.....@.....|
000000c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |@.....@.....|
000000d0 e0 16 00 00 00 00 00 e0 16 00 00 00 00 00 00 00 |@.....@.....|
000000e0 00 10 00 00 00 00 00 01 00 00 00 05 00 00 00 00 |@.....@.....|
000000f0 00 20 00 00 00 00 00 00 20 00 00 00 00 00 00 00 |@.....@.....|
00000100 00 20 00 00 00 00 00 00 31 44 00 00 00 00 00 00 |@.....@.....|
00000110 31 44 00 00 00 00 00 00 10 00 00 00 00 00 00 00 |1D.....1D.....|
00000120 01 00 00 04 00 00 00 00 70 00 00 00 00 00 00 00 |@.....@.....|
00000130 00 70 00 00 00 00 00 00 70 00 00 00 00 00 00 00 |@.....@.....|
00000140 d0 21 00 00 00 00 00 d0 21 00 00 00 00 00 00 00 |!.....!.....|
00000150 00 10 00 00 00 00 00 01 00 00 06 00 00 00 00 00 |@.....@.....|
00000160 90 9a 00 00 00 00 00 90 9a 00 00 00 00 00 00 00 |@.....@.....|
00000170 90 9a 00 00 00 00 00 30 06 00 00 00 00 00 00 00 |@.....@.....|
00000180 c8 07 00 00 00 00 00 00 10 00 00 00 00 00 00 00 |@.....@.....|
00000190 02 00 00 06 00 00 00 38 9c 00 00 00 00 00 00 00 |@.....@.....|
000001a0 38 ac 00 00 00 00 00 38 ac 00 00 00 00 00 00 00 |8.....8.....|
000001b0 f0 01 00 00 00 00 00 f0 01 00 00 00 00 00 00 00 |@.....@.....|
000001c0 08 00 00 00 00 00 00 04 00 00 00 04 00 00 00 00 |@.....@.....|
000001d0 38 03 00 00 00 00 00 38 03 00 00 00 00 00 00 00 |8.....8.....|
000001e0 38 03 00 00 00 00 00 20 00 00 00 00 00 00 00 00 |8.....8.....|
000001f0 20 00 00 00 00 00 00 08 00 00 00 00 00 00 00 00 |@.....@.....|
00000200 04 00 00 04 00 00 00 58 03 00 00 00 00 00 00 00 |X.....X.....|
00000210 58 03 00 00 00 00 00 58 03 00 00 00 00 00 00 00 |X.....X.....|
00000220 44 00 00 00 00 00 00 44 00 00 00 00 00 00 00 00 |D.....D.....|

```

#### 1.4.3.4 less

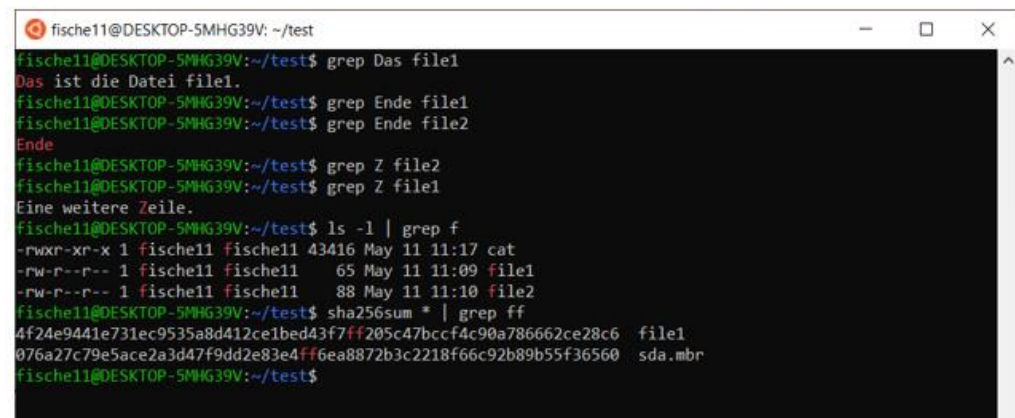
Der Befehl „less“ arbeitet nach dem Prinzip „weniger ist mehr“. Er arbeitet wie „more“, wobei hier zusätzlich hochscrollen möglich ist. Mit „q“ kann die Ausgabe von „less“ beendet werden.



```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ less file2
fische11@DESKTOP-5MHG39V:~/test$ cat file1 | less
fische11@DESKTOP-5MHG39V:~/test$ hexdump -C /bin/cat | less
fische11@DESKTOP-5MHG39V:~/test$
```

#### 1.4.3.5 grep

Der Befehl „grep“ ist die Kurzform von global regular expression print. Mit „grep“ ist zeilenweises Filtern der Ausgabe möglich.



```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ grep Das file1
Das ist die Datei file1.
fische11@DESKTOP-5MHG39V:~/test$ grep Ende file1
fische11@DESKTOP-5MHG39V:~/test$ grep Ende file2
Ende
fische11@DESKTOP-5MHG39V:~/test$ grep Z file2
fische11@DESKTOP-5MHG39V:~/test$ grep Z file1
Eine weitere Zeile.
fische11@DESKTOP-5MHG39V:~/test$ ls -l | grep f
-rwxr-xr-x 1 fische11 fische11 43416 May 11 11:17 cat
-rw-r--r-- 1 fische11 fische11 65 May 11 11:09 file1
-rw-r--r-- 1 fische11 fische11 88 May 11 11:10 file2
fische11@DESKTOP-5MHG39V:~/test$ sha256sum * | grep ff
4f24e9441e731ec9535a8d412ce1bed43f7ff205c47bccf4c90a786662ce28c6 file1
076a27c79e5ace2a3d47f9dd2e83e4ff6ea8872b3c2218f66c92b89b55f36560 sda.mbr
fische11@DESKTOP-5MHG39V:~/test$
```

#### 1.4.3.6 sort

Der Befehl „sort“ wird für das Sortieren der zeilenweisen Ausgabe genutzt. Mit dem Parameter -r kann die Reihenfolge der zeilenweisen Ausgabe umgekehrt werden.



```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ cat file1
Das ist die Datei file1.
Nur ein wenig Text.
Eine weitere Zeile.
fische11@DESKTOP-5MHG39V:~/test$ sort file1
Das ist die Datei file1.
Eine weitere Zeile.
Nur ein wenig Text.
fische11@DESKTOP-5MHG39V:~/test$ sort -r file1
Nur ein wenig Text.
Eine weitere Zeile.
Das ist die Datei file1.
fische11@DESKTOP-5MHG39V:~/test$ cat file2
Das ist die Datei file2.
Hier steht auch etwas Text.

Nach der Leere kommt das:
Ende
fische11@DESKTOP-5MHG39V:~/test$ sort file2

Das ist die Datei file2.
Ende
Hier steht auch etwas Text.
Nach der Leere kommt das:
fische11@DESKTOP-5MHG39V:~/test$
```

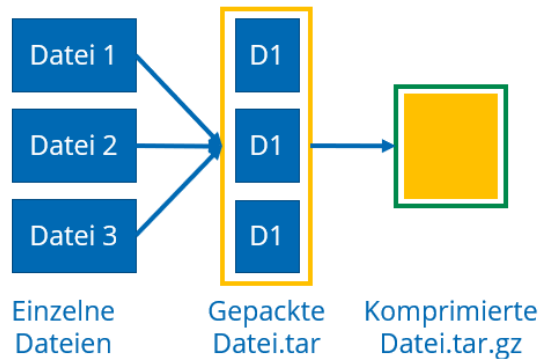
#### 1.4.3.7 wc

Der Befehl „wc“ ist die Kurzform von word count. Es können verschiedene Parameter angegeben werden. Mit -w werden Wörter, mit -l Zeilen und mit -m Zeichen gezählt. -c gibt die Größe in Bytes an und -L die maximale Zeilenlänge.

```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ wc -w file1
12 file1
fische11@DESKTOP-5MHG39V:~/test$ wc -l file1
3 file1
fische11@DESKTOP-5MHG39V:~/test$ wc -m file1
65 file1
fische11@DESKTOP-5MHG39V:~/test$ wc -c file1
65 file1
fische11@DESKTOP-5MHG39V:~/test$ wc -L file1
24 file1
fische11@DESKTOP-5MHG39V:~/test$ wc file1
 3 12 65 file1
fische11@DESKTOP-5MHG39V:~/test$
```

#### 1.4.3.8 tar

Der Befehl „tar“ ist die Kurzform von tape archiver. tar ist ein Packungsprogramm und führt Kompressionen über integrierte Kompressionsalgorithmen durch.



```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ tar -cf packed file1 file2
fische11@DESKTOP-5MHG39V:~/test$ tar -czf compressed file1 file2
fische11@DESKTOP-5MHG39V:~/test$ ls -lh packed compressed
-rw-r--r-- 1 fische11 fische11 251 May 11 12:27 compressed
-rw-r--r-- 1 fische11 fische11 10K May 11 12:27 packed
fische11@DESKTOP-5MHG39V:~/test$ tar -xvf compressed
file1
file2
fische11@DESKTOP-5MHG39V:~/test$
```

## 1.4.4 Rechteverwaltung

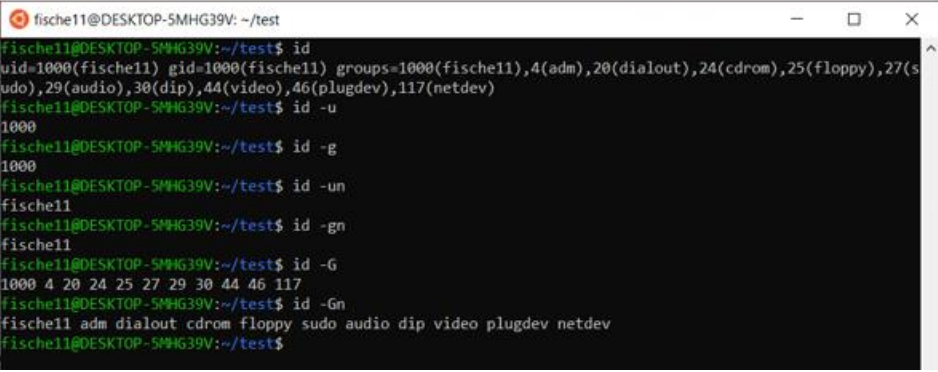
### 1.4.4.1 who

Der Befehl „who“, wie auch „whoami“, den aktuellen Nutzer an und kann alle angemeldeten Benutzer auflisten. Diese Ausgabe kann mit Parametern noch spezifiziert werden. Mit -a werden alle angezeigt, mit -u nur angemeldete Nutzer, mit -b wird noch die Bootzeit ausgegeben, mit -t die letzte Zeitänderung, mit -q die Anzahl der angemeldeten Nutzer und mit -r wird das aktuelle Runlevel mit ausgegeben.

```
LinuxMint [wird ausgeführt] - Oracle VM VirtualBox
Datei Maschine Anzeige Eingabe Geräte Hilfe
linuxmint@linuxmint-VirtualBox: ~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
linuxmint@linuxmint-VirtualBox:~$ whoami
linuxmint
linuxmint@linuxmint-VirtualBox:~$ who -u
linuxmint tty7      2022-05-11 13:26 00:04      1751 (:0)
linuxmint@linuxmint-VirtualBox:~$ who -a
Systemstart 2022-05-11 13:25
LOGIN      tty1      2022-05-11 13:25      760 id=tty1
Runlevel 5 2022-05-11 13:25
linuxmint + tty7      2022-05-11 13:26 00:04      1751 (:0)
linuxmint@linuxmint-VirtualBox:~$ who -b
Systemstart 2022-05-11 13:25
linuxmint@linuxmint-VirtualBox:~$ who -q
linuxmint
# Benutzer=1
linuxmint@linuxmint-VirtualBox:~$ who -t
linuxmint@linuxmint-VirtualBox:~$ who -r
Runlevel 5 2022-05-11 13:25
linuxmint@linuxmint-VirtualBox:~$
```

#### 1.4.4.2 id

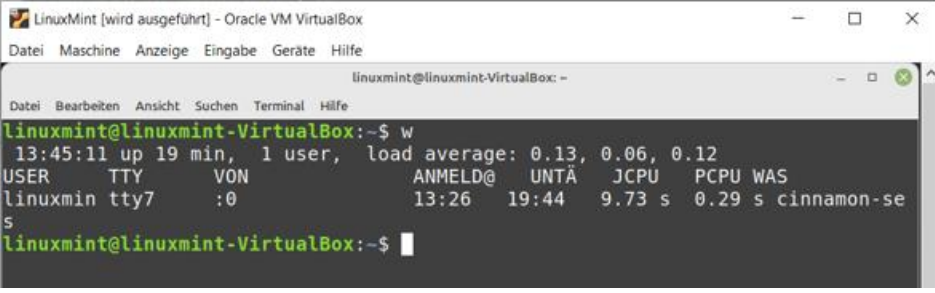
Der Befehl „id“ zeigt die User- und die Gruppen-ID an. Auch dieser Befehl kann mit Parametern erweitert werden. Mit dem Parameter -u wird nur die User-ID angezeigt, wohingegen mit -g nur die Gruppen-ID ausgegeben wird. Mit -G werden alle Gruppen-IDs angezeigt und mit -n erfolgt eine Namensauflösung der IDs.



```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ id
uid=1000(fische11) gid=1000(fische11) groups=1000(fische11),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),117(netdev)
fische11@DESKTOP-5MHG39V:~/test$ id -u
1000
fische11@DESKTOP-5MHG39V:~/test$ id -g
1000
fische11@DESKTOP-5MHG39V:~/test$ id -un
fische11
fische11@DESKTOP-5MHG39V:~/test$ id -gn
fische11
fische11@DESKTOP-5MHG39V:~/test$ id -G
1000 4 20 24 25 27 29 30 44 46 117
fische11@DESKTOP-5MHG39V:~/test$ id -Gn
fische11 adm dialout cdrom floppy sudo audio dip video plugdev netdev
fische11@DESKTOP-5MHG39V:~/test$
```

#### 1.4.4.3 w

Der Befehl „w“ zeigt aktive Benutzer an. Außerdem wird der aktuell genutzte interaktive Prozess der Nutzer angezeigt.



```
LinuxMint [wird ausgeführt] - Oracle VM VirtualBox
Datei Maschine Anzeige Eingabe Geräte Hilfe
linuxmint@linuxmint-VirtualBox: -
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
linuxmint@linuxmint-VirtualBox:~$ w
13:45:11 up 19 min, 1 user, load average: 0.13, 0.06, 0.12
USER      TTY      VON           ANMELD@   UNTÄ      JCPU    PCPU  WAS
linuxmin  tty7     :0            13:26    19:44    9.73 s  0.29 s  cinnamon-se
s
linuxmint@linuxmint-VirtualBox:~$
```

#### 1.4.4.4 adduser

Mit dem Befehl „adduser“ werden Benutzer und Benutzer zu Gruppen hinzugefügt. Mit der Option --home kann man ein vom Standard abweichendes Home-Verzeichnis definieren. --no-create-home ist die Option, damit kein Home-Verzeichnis erstellt wird. Mit der Option --quiet erfolgt keine Interaktion, sprich es werden keine möglichen Warnmeldungen ausgegeben. Mit --uid erzwingt man das Erstellen einer speziellen User-ID. Die letzte Option ist --system, mit welcher ein Systembenutzer erstellt werden kann.

```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ sudo adduser leon
Adding user `leon' ...
Adding new group `leon' (1001) ...
Adding new user `leon' (1001) with group `leon' ...
Creating home directory `/home/leon' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for leon
Enter the new value, or press ENTER for the default
    Full Name []: Leon
    Room Number []: 8-107
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n]
fische11@DESKTOP-5MHG39V:~/test$ id -u leon
1001
fische11@DESKTOP-5MHG39V:~/test$ id -g leon
1001
fische11@DESKTOP-5MHG39V:~/test$ adduser leon fische11
adduser: Only root may add a user or group to the system.
fische11@DESKTOP-5MHG39V:~/test$ sudo adduser leon fische11
Adding user `leon' to group `fische11' ...
Adding user leon to group fische11
Done.
fische11@DESKTOP-5MHG39V:~/test$ id -g leon
1001
fische11@DESKTOP-5MHG39V:~/test$ id leon
uid=1001(leon) gid=1001(leon) groups=1001(leon),1000(fische11)
fische11@DESKTOP-5MHG39V:~/test$
```

#### 1.4.4.5 addgroup

Mit dem Befehl „addgroup“ kann man eine Gruppe erstellen. Mit der Option --system wird eine Systemgruppe erstellt. Alternativ kann das Erstellen einer Gruppe bzw. Systemgruppe auch über folgende Befehle erfolgen: „adduser --group“ bzw. „adduser --group --system“

```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ sudo addgroup programmers
Adding group `programmers' (GID 1002) ...
Done.
fische11@DESKTOP-5MHG39V:~/test$ sudo adduser --group sellforce
Adding group `sellforce' (GID 1003) ...
Done.
fische11@DESKTOP-5MHG39V:~/test$
```

#### 1.4.4.6 passwd

Der Befehl „passwd“ ist die Kurzform von password und ist für die Änderung von Passwörtern zuständig. Wenn man sein eigenes Passwort ändern möchte, wird das Alte dafür benötigt. Weiterhin kann der Root Nutzer die Passwörter anderer Nutzer überschreiben.

```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ passwd
Changing password for fische11.
Current password:
New password:
Retype new password:
passwd: password updated successfully
fische11@DESKTOP-5MHG39V:~/test$ sudo passwd leon
New password:
Retype new password:
passwd: password updated successfully
fische11@DESKTOP-5MHG39V:~/test$
```

#### 1.4.4.7 chmod

Der Befehl „chmod“ ist die Kurzform von change mode bits und ist dafür zuständig, die Rechte Bits für eine Datei zu setzen. Der Befehlsaufbau lautet wie folgt: „chmod <Rechte> <Datei>“. Die Angabe der Rechte erfolgt in Zahlen oder mit den Rechtekürzeln. Wählt man den Parameter -R, so werden die gewählten Rechte rekursiv auf die Unterverzeichnisse angewandt.

```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ ls -l
total 76
-rwxr-xr-x 1 fische11 fische11 43416 May 11 11:17 cat
-rw-r--r-- 1 fische11 fische11 251 May 11 12:27 compressed
-rw-r--r-- 1 fische11 fische11 65 May 11 11:09 file1
-rw-r--r-- 1 fische11 fische11 88 May 11 11:10 file2
-rw-r--r-- 1 fische11 fische11 10240 May 11 12:27 packed
-rw-r--r-- 1 root root 512 May 11 11:15 sda.mbr
-rw-r--r-- 1 root root 4096 May 11 11:16 sdb.mbr
fische11@DESKTOP-5MHG39V:~/test$ chmod 400 packed
fische11@DESKTOP-5MHG39V:~/test$ ls -l
total 76
-rwxr-xr-x 1 fische11 fische11 43416 May 11 11:17 cat
-rw-r--r-- 1 fische11 fische11 251 May 11 12:27 compressed
-rw-r--r-- 1 fische11 fische11 65 May 11 11:09 file1
-rw-r--r-- 1 fische11 fische11 88 May 11 11:10 file2
-r----- 1 fische11 fische11 10240 May 11 12:27 packed
-rw-r--r-- 1 root root 512 May 11 11:15 sda.mbr
-rw-r--r-- 1 root root 4096 May 11 11:16 sdb.mbr
fische11@DESKTOP-5MHG39V:~/test$ chmod 660 compressed
fische11@DESKTOP-5MHG39V:~/test$ ls -l
total 76
-rwxr-xr-x 1 fische11 fische11 43416 May 11 11:17 cat
-rw-rw---- 1 fische11 fische11 251 May 11 12:27 compressed
-rw-r--r-- 1 fische11 fische11 65 May 11 11:09 file1
-rw-r--r-- 1 fische11 fische11 88 May 11 11:10 file2
-r----- 1 fische11 fische11 10240 May 11 12:27 packed
-rw-r--r-- 1 root root 512 May 11 11:15 sda.mbr
-rw-r--r-- 1 root root 4096 May 11 11:16 sdb.mbr
fische11@DESKTOP-5MHG39V:~/test$ chmod a+r compressed
fische11@DESKTOP-5MHG39V:~/test$ ls -l
total 76
-rwxr-xr-x 1 fische11 fische11 43416 May 11 11:17 cat
-rw-rw---- 1 fische11 fische11 251 May 11 12:27 compressed
-rw-r--r-- 1 fische11 fische11 65 May 11 11:09 file1
-rw-r--r-- 1 fische11 fische11 88 May 11 11:10 file2
-r----- 1 fische11 fische11 10240 May 11 12:27 packed
-rw-r--r-- 1 root root 512 May 11 11:15 sda.mbr
-rw-r--r-- 1 root root 4096 May 11 11:16 sdb.mbr
fische11@DESKTOP-5MHG39V:~/test$ chmod ug-w compressed
fische11@DESKTOP-5MHG39V:~/test$ ls -l
total 76
-rwxr-xr-x 1 fische11 fische11 43416 May 11 11:17 cat
-r--r--r-- 1 fische11 fische11 251 May 11 12:27 compressed
-rw-r--r-- 1 fische11 fische11 65 May 11 11:09 file1
-rw-r--r-- 1 fische11 fische11 88 May 11 11:10 file2
-r----- 1 fische11 fische11 10240 May 11 12:27 packed
-rw-r--r-- 1 root root 512 May 11 11:15 sda.mbr
-rw-r--r-- 1 root root 4096 May 11 11:16 sdb.mbr
```

#### 1.4.4.8 chown

Der Befehl „chown“ ist die Kurzform von change owner. Mit diesem Befehl kann man den Besitzer und die Gruppe ändern. Der Befehlsaufbau lautet wie folgt: „chown <Besitzer>:<Gruppe> <Datei>“. Mit dem Parameter -R kann der Besitzer rekursiv auf die Unterverzeichnisse angewandt werden.

```

fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ ls -l
total 76
-rwxr-xr-x 1 fische11 fische11 43416 May 11 11:17 cat
-r--r--r-- 1 fische11 fische11 251 May 11 12:27 compressed
-rw-r--r-- 1 fische11 fische11 65 May 11 11:09 file1
-rw-r--r-- 1 fische11 fische11 88 May 11 11:10 file2
-r----- 1 fische11 fische11 10240 May 11 12:27 packed
-rw-r--r-- 1 root root 512 May 11 11:15 sda.mbr
-rw-r--r-- 1 root root 4096 May 11 11:16 sdb.mbr
fische11@DESKTOP-5MHG39V:~/test$ chown fische11:sudo packed
fische11@DESKTOP-5MHG39V:~/test$ ls -l
total 76
-rwxr-xr-x 1 fische11 fische11 43416 May 11 11:17 cat
-r--r--r-- 1 fische11 fische11 251 May 11 12:27 compressed
-rw-r--r-- 1 fische11 fische11 65 May 11 11:09 file1
-rw-r--r-- 1 fische11 fische11 88 May 11 11:10 file2
-r----- 1 fische11 sudo 10240 May 11 12:27 packed
-rw-r--r-- 1 root root 512 May 11 11:15 sda.mbr
-rw-r--r-- 1 root root 4096 May 11 11:16 sdb.mbr
fische11@DESKTOP-5MHG39V:~/test$ sudo chown fische11:fische11 *
[sudo] password for fische11:
fische11@DESKTOP-5MHG39V:~/test$ ls -l
total 76
-rwxr-xr-x 1 fische11 fische11 43416 May 11 11:17 cat
-r--r--r-- 1 fische11 fische11 251 May 11 12:27 compressed
-rw-r--r-- 1 fische11 fische11 65 May 11 11:09 file1
-rw-r--r-- 1 fische11 fische11 88 May 11 11:10 file2
-r----- 1 fische11 fische11 10240 May 11 12:27 packed
-rw-r--r-- 1 fische11 fische11 512 May 11 11:15 sda.mbr
-rw-r--r-- 1 fische11 fische11 4096 May 11 11:16 sdb.mbr
fische11@DESKTOP-5MHG39V:~/test$

```

#### 1.4.4.9 su

Der Befehl „su“ steht für substitute user. Mit dem Aufruf „su <Benutzer>“ erfolgt die Anmeldung als anderer Nutzer. Gibt man hingegen nur su an, erfolgt die Anmeldung als root. Mit dem Befehl „su <Benutzer> -c <Befehl>“ wird der Befehl als anderer Nutzer ausgeführt. Für jede Nutzungsvariante von „su“ wird immer das Passwort des substituierten Nutzers benötigt.

```

fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ su leon
Password:
leon@DESKTOP-5MHG39V:/home/fische11/test$ whoami
leon
leon@DESKTOP-5MHG39V:/home/fische11/test$ exit
fische11@DESKTOP-5MHG39V:~/test$ whoami
fische11
fische11@DESKTOP-5MHG39V:~/test$ su leon -c whoami
Password:
leon
fische11@DESKTOP-5MHG39V:~/test$ whoami
fische11
fische11@DESKTOP-5MHG39V:~/test$

```

#### 1.4.4.10 sudo

Der Befehl „sudo“ steht für substitute user do oder auch für super user do. Damit wird ein Befehl als anderer Nutzer („sudo -u <Benutzer> <Befehl>“) oder als root ausgeführt („sudo <Befehl>“), wobei das eigene Passwort benötigt wird und der Nutzer ein Mitglied der sudo-Gruppe sein muss.

```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ sudo whoami
root
fische11@DESKTOP-5MHG39V:~/test$ sudo -u leon whoami
leon
fische11@DESKTOP-5MHG39V:~/test$
```

#### 1.4.4.11 *logout / exit*

Der Befehl „logout“ funktioniert nur in der Loginshell. Eine nicht-graphische Shell öffnet man beispielsweise über `strg+alt+F1`, .... `F6`. „lgout“ führt die Abmeldung durch. In der normalen Shell wird `exit` für das Abmelden genutzt. Alternativ kann man auch das Tastenkürzel `strg+d` (`detach`) zum Abmelden verwenden.

```
LinuxMint [wird ausgeführt] - Oracle VM VirtualBox
Datei Maschine Anzeige Eingabe Geräte Hilfe

Linux Mint 20.3 Una linuxmint-VirtualBox tty1
linuxmint-VirtualBox login: linuxmint
Password:
Last login: Wed May 11 14:56:37 CEST 2022 on tty1
linuxmint@linuxmint-VirtualBox:~$ logout_
```

#### 1.4.4.12 *shutdown*

Der Befehl „shutdown“ wird genutzt, um das System herunterzufahren. Hierbei wird noch eine Minute gewartet, bevor das System heruntergefahren wird. Soll das System sofort heruntergefahren werden, nutzt man „shutdown now“.

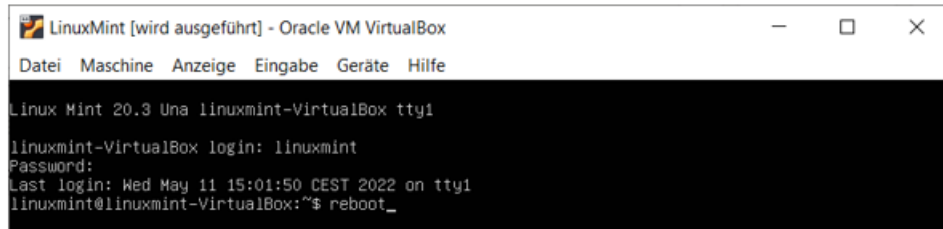
```
LinuxMint [wird ausgeführt] - Oracle VM VirtualBox
Datei Maschine Anzeige Eingabe Geräte Hilfe

Linux Mint 20.3 Una linuxmint-VirtualBox tty1
linuxmint-VirtualBox login: linuxmint
Password:
Last login: Wed May 11 14:56:52 CEST 2022 on tty1
linuxmint@linuxmint-VirtualBox:~$ shutdown
Shutdown scheduled for Wed 2022-05-11 15:03:25 CEST, use 'shutdown -c' to cancel.
linuxmint@linuxmint-VirtualBox:~$ _
```



#### 1.4.4.13 reboot

Der Befehl „reboot“ startet das System neu. Dieser Befehl wird im Gegensatz zu „shutdown“ direkt ohne Verzögerung ausgeführt.



```
LinuxMint [wird ausgeführt] - Oracle VM VirtualBox
Datei Maschine Anzeige Eingabe Geräte Hilfe
Linux Mint 20.3 Una linuxmint-VirtualBox tty1
linuxmint-VirtualBox login: linuxmint
Password:
Last login: Wed May 11 15:01:50 CEST 2022 on tty1
linuxmint@linuxmint-VirtualBox:~$ reboot_
```

#### 1.4.5 Zusammenfassung

Sie kennen die Aufgaben einer Bash und alternative Shells. Heute habe Sie folgende Shell- Anwendungen kennengelernt: ls, cd, pwd, cp, mv, touch, rm, mkdir, rmdir, ln, find, cat, hexdump, more, less, grep, sort, wc, tar, who, id, w, adduser, addgroup, passwd, chmod, chown, su, sudo, logout, exit, shutdown und reboot.

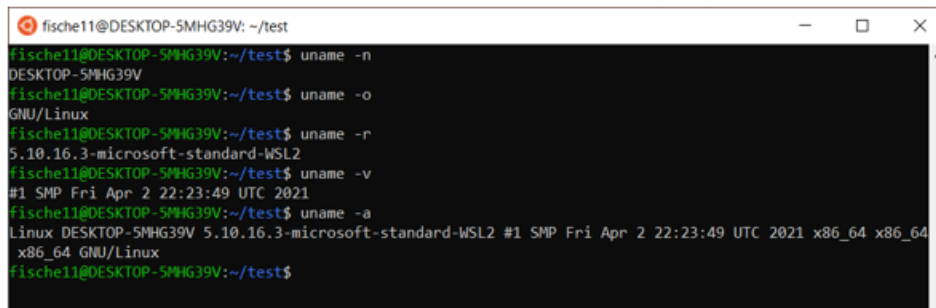


## 1.5 Linux Anwendungen im Terminal 2

### 1.5.1 Systemüberblick

#### 1.5.1.1 `uname`

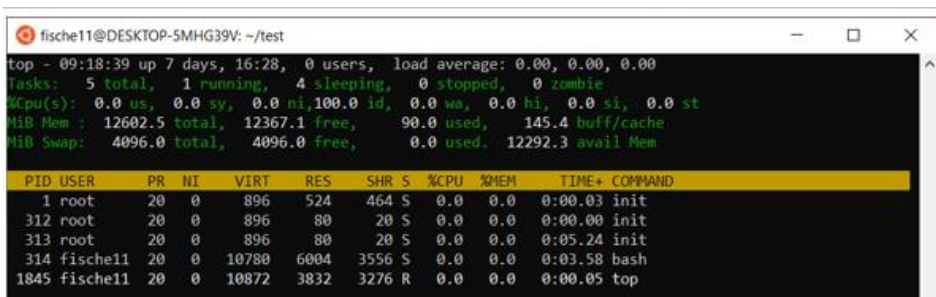
Der Befehl „`uname`“ gibt allgemeine Informationen über das Betriebssystem aus. Die Ausgabe kann über verschiedene Parameter angepasst werden. `-a` gibt alles aus, `-n` den Netzwerk- bzw. Hostnamen, `-o` den Betriebssystemnamen, `-r` den Kernel release und `-v` die Kernel Version.



```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ uname -n
DESKTOP-5MHG39V
fische11@DESKTOP-5MHG39V:~/test$ uname -o
GNU/Linux
fische11@DESKTOP-5MHG39V:~/test$ uname -r
5.10.16.3-microsoft-standard-WSL2
fische11@DESKTOP-5MHG39V:~/test$ uname -v
#1 SMP Fri Apr 2 22:23:49 UTC 2021
fische11@DESKTOP-5MHG39V:~/test$ uname -a
Linux DESKTOP-5MHG39V 5.10.16.3-microsoft-standard-WSL2 #1 SMP Fri Apr 2 22:23:49 UTC 2021 x86_64 x86_64
x86_64 GNU/Linux
fische11@DESKTOP-5MHG39V:~/test$
```

#### 1.5.1.2 `top`

Der Parameter „`top`“ zeigt den Prozessmonitor an. „`s`“ ist das Updateintervall, mit „`f`“ werden die angezeigten Felder eingestellt, „`E`“ schaltet in Si-Einheiten um, „`Z`“ ist für die Farbeinstellung und „`q`“ beendet den Prozessmonitor.



```
fische11@DESKTOP-5MHG39V: ~/test
top - 09:18:39 up 7 days, 16:28, 0 users, load average: 0.00, 0.00, 0.00
Tasks: 5 total, 1 running, 4 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 12602.5 total, 12367.1 free, 90.0 used, 145.4 buff/cache
MiB Swap: 4096.0 total, 4096.0 free, 0.0 used, 12292.3 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
  1   root        0   0    896    524   464  S   0.0   0.0   0:00.03  init
 312  root        20   0    896     80    20  S   0.0   0.0   0:00.00  init
 313  root        20   0    896     80    20  S   0.0   0.0   0:05.24  init
 314  fische11    20   0  10780  6004  3556  S   0.0   0.0   0:03.58  bash
1845  fische11    20   0  10872  3822  3276  R   0.0   0.0   0:00.05  top
```

#### 1.5.1.3 `ps`

„`ps`“ steht für process snapshot. Mit `-e` werden alle Prozesse angezeigt, `-f` liefert eine vollständige Anzeige, `-l` ein langes Format, `-a` gibt alle Prozesse aus, außer die vom ausführenden Terminal gestartet wurden, `-x` zeigt nur eigene Prozesse an und `-u` liefert eine benutzerorientierte Ausgabe.

```

fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ ps -e
PID TTY          TIME CMD
  1 ?            00:00:00 init
 312 ?            00:00:00 init
 313 ?            00:00:05 init
 314 pts/0       00:00:03 bash
1941 pts/0       00:00:00 ps
fische11@DESKTOP-5MHG39V:~/test$ ps -elf
F S UID          PID  PPID  C PRI  NI ADDR SZ WCHAN  STATE  TIME  TTY          TIME CMD
4 S root           1    0    0  80   0 - 224 -  May04 ?    00:00:00 /init
5 S root          312   1    0  80   0 - 224 -  May04 ?    00:00:00 /init
1 R root          313  312   0  80   0 - 224 -  May04 ?    00:00:05 /init
4 S fische11     314  313   0  80   0 - 2695 do_wai May04 pts/0  00:00:03 -bash
0 R fische11     1942 314   0  80   0 - 2654 -    09:30 pts/0  00:00:00 ps -elf
fische11@DESKTOP-5MHG39V:~/test$ ps -aux
USER          PID  %CPU  %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.0   896   524 ?        S1   May04   0:00 /init
root          312  0.0  0.0   896    80 ?        Ss  May04   0:00 /init
root          313  0.0  0.0   896    80 ?        S   May04   0:05 /init
fische11     314  0.0  0.0 10780 6004 pts/0    Ss  May04   0:03 -bash
fische11     1943  0.0  0.0 10856 3408 pts/0    R+  09:30   0:00 ps -aux
fische11@DESKTOP-5MHG39V:~/test$ ps -eu
USER          PID  %CPU  %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
fische11     314  0.0  0.0 10780 6004 pts/0    Ss  May04   0:03 -bash HOSTTYPE=x86_64 LANG=C.UTF-8 PATH=
fische11     1944  0.0  0.0 10616 3256 pts/0    R+  09:31   0:00 ps -eu SHELL=/bin/bash WSL_DISTRO_NAME=
fische11@DESKTOP-5MHG39V:~/test$ ps -a
PID TTY          TIME CMD
1945 pts/0       00:00:00 ps
fische11@DESKTOP-5MHG39V:~/test$ ps -x
PID TTY          STAT TIME COMMAND
 314 pts/0       Ss   0:03 -bash
1946 pts/0       R+   0:00 ps -x
fische11@DESKTOP-5MHG39V:~/test$ ps -xf
PID TTY          STAT TIME COMMAND
 314 pts/0       Ss   0:03 -bash
1947 pts/0       R+   0:00 \ ps -xf
fische11@DESKTOP-5MHG39V:~/test$ ps -elf | grep top
0 S fische11 2015 1950 0 80 0 - 2719 -    09:33 pts/1  00:00:00 top
0 S fische11 2017 314 0 80 0 - 2041 pipe_r 09:33 pts/0  00:00:00 grep --color=auto top
fische11@DESKTOP-5MHG39V:~/test$

```

### 1.5.1.4 kill

Der Befehl „kill“ sendet ein Signal an einen Prozess. -l listet alle Signale auf. -2 ist ein Interrupt, -3 eine Aufforderung für das Beenden des Prozesses und -9 ein erzwungenes Beenden.

```

fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ kill -l
 1) SIGHUP      2) SIGINT     3) SIGQUIT   4) SIGILL    5) SIGTRAP
 6) SIGABRT    7) SIGBUS    8) SIGFPE    9) SIGKILL   10) SIGUSR1
11) SIGSEGV   12) SIGUSR2  13) SIGPIPE  14) SIGALRM  15) SIGTERM
16) SIGSTKFLT 17) SIGCHLD  18) SIGCONT  19) SIGSTOP  20) SIGTSTP
21) SIGTTIN   22) SIGTTOU  23) SIGURG   24) SIGXCPU  25) SIGXFSZ
26) SIGVTALRM 27) SIGPROF  28) SIGWINCH 29) SIGIO    30) SIGPWR
31) SIGSYS    34) SIGRTMIN 35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7
58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
63) SIGRTMAX-1 64) SIGRTMAX
fische11@DESKTOP-5MHG39V:~/test$ ps -e | grep top
2050 pts/1    00:00:00 top
fische11@DESKTOP-5MHG39V:~/test$ kill -3 2050
fische11@DESKTOP-5MHG39V:~/test$

```

### 1.5.1.5 date

„date“ gibt die Systemzeit aus und man kann sie hierüber einstellen. Mit „+“ kann man eine gewünschte Formatierung angeben, mit -d das übergebene Datum ausgeben bzw. formatieren, mit -s eine Zeit festlegen und mit -u die UTC-Zeit ausgeben lassen.

```

fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ date
Thu May 12 09:49:12 CEST 2022
fische11@DESKTOP-5MHG39V:~/test$ date +"%d.%m.%Y"
12.05.2022
fische11@DESKTOP-5MHG39V:~/test$ date +"%d.%m.%Y %H:%M:%S"
12.05.2022 09:50:10
fische11@DESKTOP-5MHG39V:~/test$ date +"Tag im Jahr: %j"
Tag im Jahr: 132
fische11@DESKTOP-5MHG39V:~/test$ date +"Wochentag: %w"
Wochentag: 4
fische11@DESKTOP-5MHG39V:~/test$ date +"Wochennummer: %W"
Wochennummer: 19
fische11@DESKTOP-5MHG39V:~/test$ date +"Zeitversatz zu UTC: %:::z"
Zeitversatz zu UTC: +02
fische11@DESKTOP-5MHG39V:~/test$ date +"Zeitzone: %Z"
Zeitzone: CEST
fische11@DESKTOP-5MHG39V:~/test$ date +"Wochentag: %A"
Wochentag: Thursday
fische11@DESKTOP-5MHG39V:~/test$ date +"Wochentag-Kürzel: %a"
Wochentag-Kürzel: Thu
fische11@DESKTOP-5MHG39V:~/test$ date +"Monat: %b"
Monat: May
fische11@DESKTOP-5MHG39V:~/test$ date -d "11/09/1989" +"Die deutsche Mauer fiel an einem: %A"
Die deutsche Mauer fiel an einem: Thursday
fische11@DESKTOP-5MHG39V:~/test$ date -d "10/03/2022" +"Der Tag der deutschen Wiedervereinigung fällt di
eses Jahr auf einem: %A"
Der Tag der deutschen Wiedervereinigung fällt dieses Jahr auf einem: Monday
fische11@DESKTOP-5MHG39V:~/test$ TZ="America/NewYork" date +"In New York ist es jetzt: %H:%M"
In New York ist es jetzt: 08:04
fische11@DESKTOP-5MHG39V:~/test$ TZ="Europe/Paris" date +"In Paris ist es jetzt: %H:%M"
In Paris ist es jetzt: 10:05
fische11@DESKTOP-5MHG39V:~/test$

```

### 1.5.1.6 lsusb

„lsusb“ ist die Kurzform von list USB. Mit diesem Befehl werden alle USB-Anschlüsse, -Geräte und detaillierte -Informationen aufgelistet.

```

LinuxMint [wird ausgeführt] - Oracle VM VirtualBox
Datei Maschine Anzeige Eingabe Geräte Hilfe

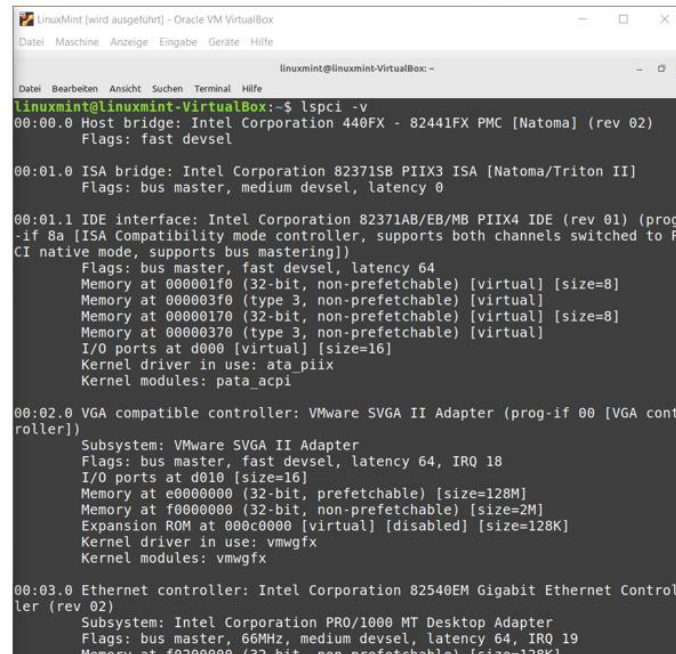
linuxmint@linuxmint-VirtualBox: ~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
linuxmint@linuxmint-VirtualBox:~$ lsusb
Bus 001 Device 004: ID 10c4:ea60 Silicon Labs CP210x UART Bridge
Bus 001 Device 002: ID 80ee:0021 VirtualBox USB Tablet
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
linuxmint@linuxmint-VirtualBox:~$ lsusb -vd 10c4:ea60

Bus 001 Device 004: ID 10c4:ea60 Silicon Labs CP210x UART Bridge
Couldn't open device, some information will be missing
Device Descriptor:
  bLength                18
  bDescriptorType        1
  bcdUSB                  2.00
  bDeviceClass            0
  bDeviceSubClass        0
  bDeviceProtocol        0
  bMaxPacketSize0       64
  idVendor                0x10c4 Silicon Labs
  idProduct              0xea60 CP210x UART Bridge
  bcdDevice              1.00
  iManufacturer          1
  iProduct               2
  iSerial                3
  bNumConfigurations     1
Configuration Descriptor:
  bLength                9
  bDescriptorType        2
  wTotalLength           0x0020
  bNumInterfaces         1
  bConfigurationValue    1
  iConfiguration        0
  bmAttributes           0x80
    (Bus Powered)
  MaxPower               100mA
Interface Descriptor:
  bLength                9

```

### 1.5.1.7 lspci

Der Befehl „lspci“ ist die Kurzform von list PCI. Mit diesem Befehl werden alle PCI-Geräte, -Anschlüsse und detaillierte -Informationen ausgegeben.



```
LinuxMint [wird ausgeführt] - Oracle VM VirtualBox
Datei Maschine Anzeige Eingabe Geräte Hilfe

linuxmint@linuxmint-VirtualBox: ~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
linuxmint@linuxmint-VirtualBox:~$ lspci -v
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
Flags: fast devsel

00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
Flags: bus master, medium devsel, latency 0

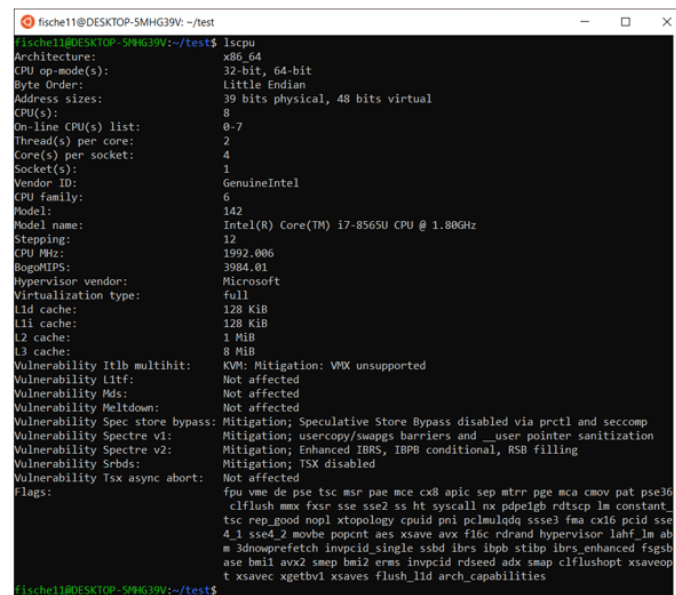
00:01.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01) (prog
-if 8a [ISA Compatibility mode controller, supports both channels switched to P
CI native mode, supports bus mastering])
Flags: bus master, fast devsel, latency 64
Memory at 000001f0 (32-bit, non-prefetchable) [virtual] [size=8]
Memory at 000003f0 (type 3, non-prefetchable) [virtual]
Memory at 00000170 (32-bit, non-prefetchable) [virtual] [size=8]
Memory at 00000370 (type 3, non-prefetchable) [virtual]
I/O ports at d000 [virtual] [size=16]
Kernel driver in use: ata_piix
Kernel modules: pata_acpi

00:02.0 VGA compatible controller: VMware SVGA II Adapter (prog-if 00 [VGA cont
roller])
Subsystem: VMware SVGA II Adapter
Flags: bus master, fast devsel, latency 64, IRQ 18
I/O ports at d010 [size=16]
Memory at e0000000 (32-bit, prefetchable) [size=128M]
Memory at f0000000 (32-bit, non-prefetchable) [size=2M]
Expansion ROM at 000c0000 [virtual] [disabled] [size=128K]
Kernel driver in use: vmwgfx
Kernel modules: vmwgfx

00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Control
ler (rev 02)
Subsystem: Intel Corporation PRO/1000 MT Desktop Adapter
Flags: bus master, 66MHz, medium devsel, latency 64, IRQ 19
Memory at f0200000 (32-bit, non-prefetchable) [size=128K]
```

### 1.5.1.8 lscpu

„lscpu“ ist die Kurzform von list CPU. Mit diesem Befehl können die CPUs aufgelistet, die CPU-Hardware-Schwachstellen angezeigt und detaillierte CPU-Informationen ausgegeben werden.

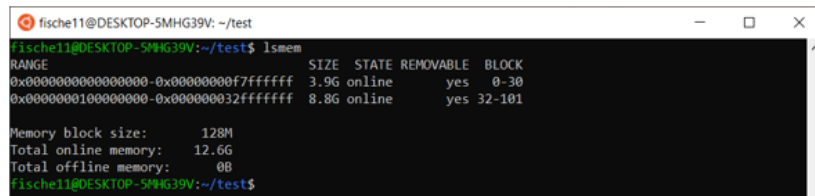


```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:             Little Endian
Address sizes:          39 bits physical, 48 bits virtual
CPU(s):                 8
On-line CPU(s) list:   0-7
Thread(s) per core:    2
Core(s) per socket:    4
Socket(s):              1
Vendor ID:              GenuineIntel
CPU family:             6
Model:                  142
Model name:             Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz
Stepping:               12
CPU MHz:                1992.006
BogoMIPS:               3984.01
Hypervisor vendor:     Microsoft
Virtualization type:   full
L1d cache:              128 KiB
L1i cache:              128 KiB
L2 cache:                1 MiB
L3 cache:                8 MiB
Vulnerability Itlb multihit: KVM: Mitigation: VMX unsupported
Vulnerability L1tf:      Not affected
Vulnerability Mds:       Not affected
Vulnerability Meltdown: Not affected
Vulnerability Spec store bypass: Mitigation; Speculative Store Bypass disabled via prctl and seccomp
Vulnerability Spectre v1: Mitigation; usercopy/swapgs barriers and __user pointer sanitization
Vulnerability Spectre v2: Mitigation; Enhanced IBRS, IBPB conditional, RSB filling
Vulnerability Srbds:     Mitigation; TSX disabled
Vulnerability Tsx async abort: Not affected
Flags:                   fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36
                        clflush mmx fxsr sse sse2 ss ht syscall nx pdpe1gb rdtscp lm constant_
                        tsc rep_good nopl xtopology cpuid pni pclmulqdq sse3 fma cx16 pcid sse
                        4_1 sse4_2 movbe popcnt aes xsave avx f16c rdrand hypervisor lahf_lm ab
                        m 3dnowprefetch invpcid_single srbds ibrs lbrs lpbp stibp lbrs_enhanced fsgs
                        bse bti1 avx2 smap bmi2 erms invpcid rdseed adx smap clflushopt xsaveso
                        p t xsavec xgetbv1 xsaves flush_l1d arch_capabilities

fische11@DESKTOP-5MHG39V:~/test$
```

### 1.5.1.9 lsmem

Der Befehl „lsmem“ steht für list memory. Mit diesem Befehl kann der RAM-Speicher aufgelistet und detaillierte RAM-Informationen ausgegeben werden.



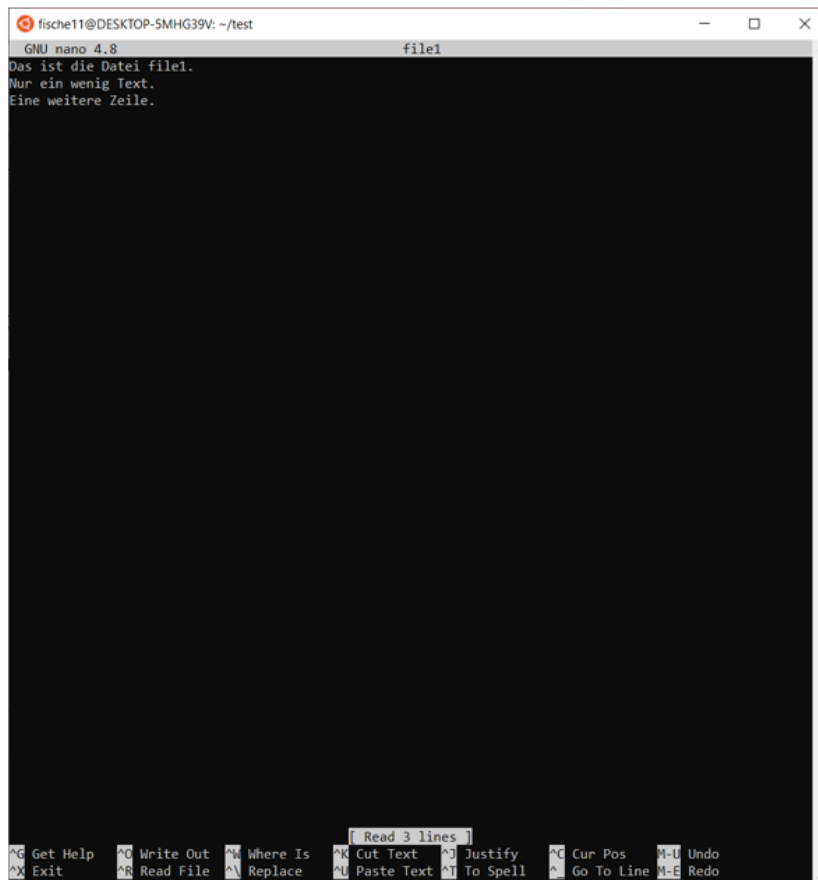
```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ lsmem
RANGE                SIZE STATE REMOVABLE BLOCK
0x0000000000000000-0x00000000f7ffffff 3.9G online    yes  0-30
0x00000000100000000-0x0000000032ffffff 8.8G online    yes 32-101

Memory block size:    128M
Total online memory:  12.6G
Total offline memory: 0B
fische11@DESKTOP-5MHG39V:~/test$
```

## 1.5.2 Konsoleneditor

### 1.5.2.1 nano

Der Editor nano ist ein einsteigerfreundlicher Editor. Die Bedienung ist leicht verständlich. Eingaben können mit Strg+o gespeichert werden. Mit Strg+w kann man suchen und mit Strg+x den Editor beenden.

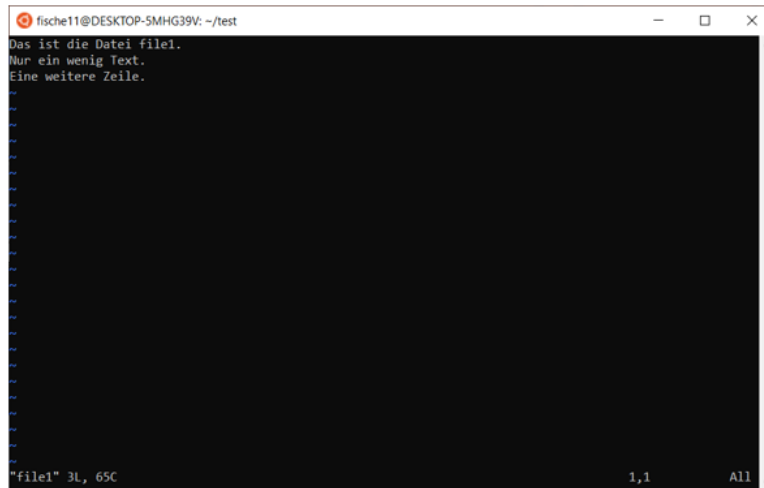


```
fische11@DESKTOP-5MHG39V: ~/test
GNU nano 4.8 file1
Das ist die Datei file1.
Nur ein wenig Text.
Eine weitere Zeile.

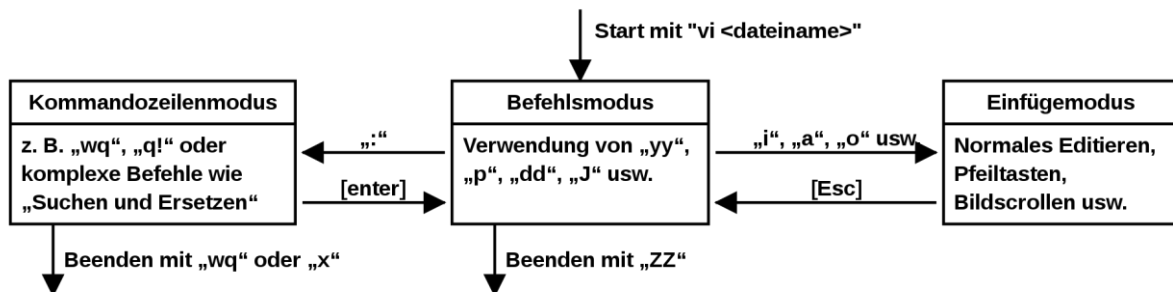
Read 3 lines
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text    ^J Justify    ^C Cur Pos   ^-U Undo
^X Exit      ^R Read File  ^M Replace   ^P Paste Text ^T To Spell  ^G Go To Line ^-E Redo
```

### 1.5.2.2 vi

vi ist die Kurzform von visual. Dieser Editor ist sehr ressourcensparend und fast immer installiert. Er ist auch auf einigen IoT-Geräten zu finden und ist aufgrund seiner vielen Befehle sehr flexibel.



Der vi Editor hat drei verschiedene Modi:



### 1.5.2.3 vim

vim ist die Kurzform von vi improved. Dieser Editor ist eine Weiterentwicklung von vi, weshalb er die gleiche Funktionsweise hat. Zusätzlich besitzt vim weitere Features wie Syntaxhighlighting, er ist voll scriptfähig und besitzt mehr Modi.

```
fische11@DESKTOP-5MHG39V: ~/test
Das ist die Datei file1.
Nur ein wenig Text.
Eine weitere Zeile.

"file1" 3l, 69c                               2,1      All
```

#### 1.5.2.4 emacs

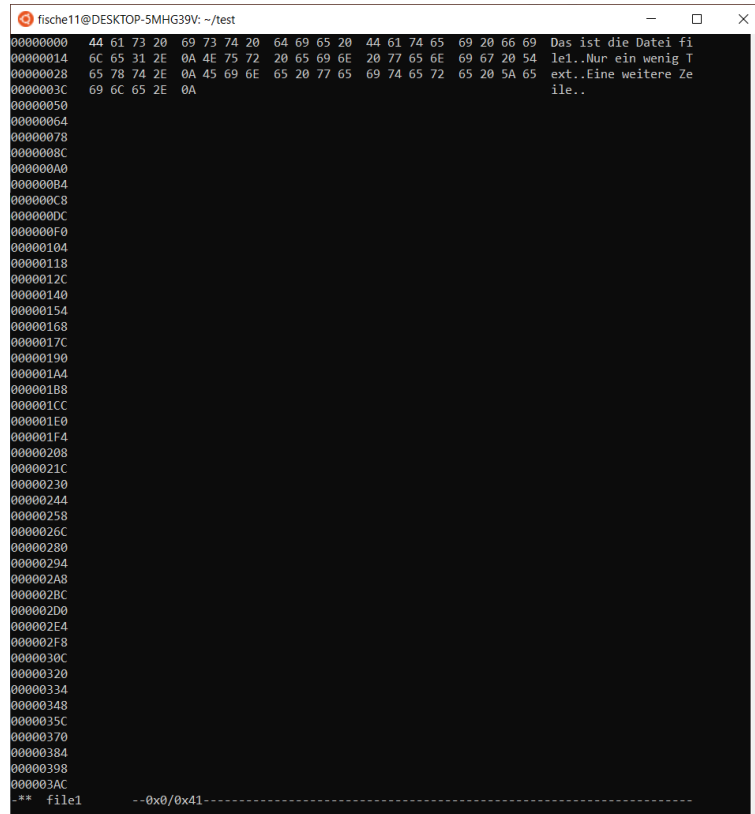
emacs ist die Kurzform von editor macros. Dieser Editor ist vergleichbar mit vim, da sie eine ähnliche Funktionsweise besitzen. Auch emacs hat verschiedene Betriebsmodi, ist voll scriptfähig und führt Syntaxhighlighting durch. Ob man eher vim oder emacs benutzt, ist weiterhin eine Streitfrage in der Linux-Welt.

```
fische11@DESKTOP-5MHG39V: ~/test
File Edit Options Buffers Tools Help
Das ist die Datei file1.
Nur ein wenig Text.
Eine weitere Zeile.

GNU Emacs 28.1.1  (Fundamental)
For information about GNU Emacs and the GNU system, type C-h C-a.
```

### 1.5.2.5 hexedit

hexedit ist die Kurzform von hex editor. Dieser Editor ist für die Bearbeitung von Bytes in Dateien zuständig.



```
fische11@DESKTOP-5MHG39V: ~/test
00000000 44 61 73 20 69 73 74 20 64 69 65 20 44 61 74 65 69 20 66 69  Das ist die Datei fi
00000014 6C 65 31 2E 0A 4E 75 72 20 65 69 6E 20 77 65 6E 69 67 20 54  le1..Nur ein wenig T
00000028 65 78 74 2E 0A 45 69 6E 65 20 77 65 69 74 65 72 65 20 5A 65  ext..Eine weitere Ze
0000003C 69 6C 65 2E 0A
00000050
00000064
00000078
0000008C
000000A0
000000B4
000000C8
000000DC
000000F0
00000104
00000118
0000012C
00000140
00000154
00000168
0000017C
00000190
000001A4
000001B8
000001CC
000001E0
000001F4
00000208
0000021C
00000230
00000244
00000258
0000026C
00000280
00000294
000002A8
000002BC
000002D0
000002E4
000002F8
0000030C
00000320
00000334
00000348
0000035C
00000370
00000384
00000398
000003AC
-#* file1 --0x0/0x41-----
```

## 1.5.3 Datenträger

### 1.5.3.1 gdisk

Mit dem Befehl „gdisk“ können Festplatten partitioniert werden. Er ist der Nachfolger von „fdisk“ und unterstützt die Partitionierungen MBR, GPT, BSD und APM. Für die Ausführung dieses Befehls benötigt man root-Rechte.



```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ sudo gdisk /dev/sda
GPT fdisk (gdisk) version 1.0.5

Partition table scan:
  MBR: not present
  BSD: not present
  APM: not present
  GPT: not present

Creating new GPT entries in memory.

Command (? for help): h
b      back up GPT data to a file
c      change a partition's name
d      delete a partition
i      show detailed information on a partition
l      list known partition types
n      add a new partition
o      create a new empty GUID partition table (GPT)
p      print the partition table
q      quit without saving changes
r      recovery and transformation options (experts only)
s      sort partitions
t      change a partition's type code
v      verify disk
w      write table to disk and exit
x      extra functionality (experts only)
?      print this menu

Command (? for help): q
fische11@DESKTOP-5MHG39V:~/test$
```

### 1.5.3.2 mkfs

„mkfs“ steht für make file system und ist für die Formatierung von Partitionen zuständig. Dabei gibt es ein Programm für jedes Filesystem:

- mkfs.ext4 (für ext4, Ubuntu)
- mkfs.ntfs (für NTFS, Windows)
- mkfs.xfs (für xfs, RedHat)
- mkfs.msdos (für fat16, MSDOS)
- mkfs.fat (für fat32, USB-Sticks)
- mkfs.vfat (für vfat, Windows95)

```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ sudo mkfs.ext4 /dev/loop1p1
mke2fs 1.45.5 (07-Jan-2020)
Discarding device blocks: done
Creating filesystem with 7931 4k blocks and 7936 inodes

Allocating group tables: done
Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done

fische11@DESKTOP-5MHG39V:~/test$
```

### 1.5.3.3 mount

Mit dem Befehl „mount“ können Dateisysteme eingebunden werden. Hierfür muss der angegebene Ordner existieren. Der Befehlsaufbau ist wie folgt: „mount -t <Type> <Gerät> <Ordner>“, wobei der Typ meist automatisch erkannt wird. Aufgrund dessen reicht „mount <Gerät> <Ordner>“. Wenn das Gerät in der fstab eingetragen ist, reicht „mount <Ordner>“.

```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ mkdir backup
fische11@DESKTOP-5MHG39V:~/test$ sudo mount /dev/loop1p1 backup
fische11@DESKTOP-5MHG39V:~/test$
```

### 1.5.3.4 unmount

Der Befehl „umount“ wird verwendet, um ein Filesystem auszubinden. Der Befehlsaufbau ist „umount <Ordner>“.

```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ mkdir backup
fische11@DESKTOP-5MHG39V:~/test$ sudo mount /dev/loop1p1 backup
fische11@DESKTOP-5MHG39V:~/test$ sudo umount backup
fische11@DESKTOP-5MHG39V:~/test$ sudo umount backup
umount: backup: not mounted.
fische11@DESKTOP-5MHG39V:~/test$
```

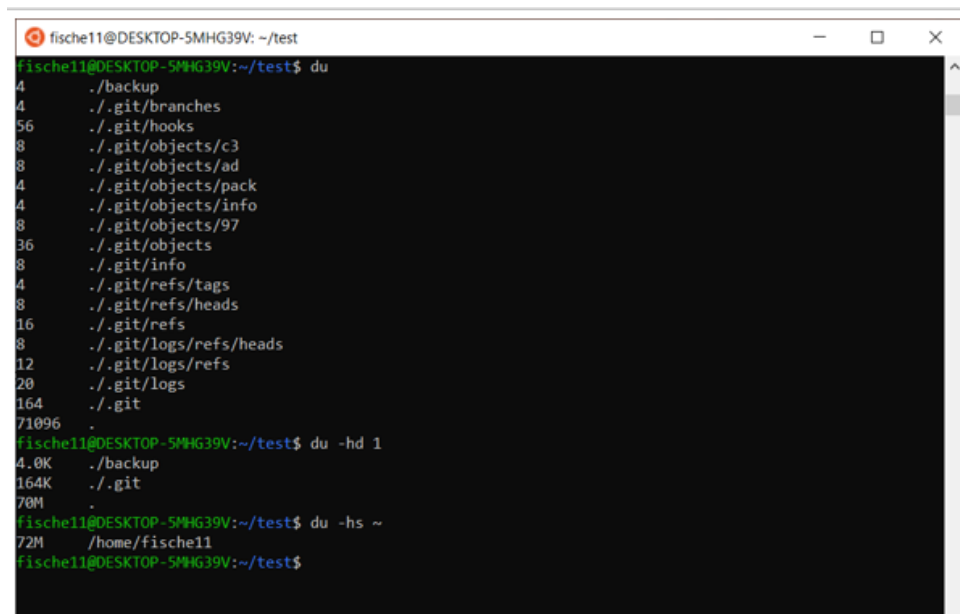
### 1.5.3.5 df

„df“ ist die Kurzform von disk free. Hiermit kann die Speichergröße von Filesystemen angezeigt werden. Mit -h wird die Größe in einer menschenlesbaren Form ausgegeben.

```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sdb        263174212  1896156 247839900   1% /
tmpfs           6452476      0 6452476   0% /mnt/wsl
tools          499448504 123221896 376226608  25% /init
none            6450392      0 6450392   0% /dev
none            6452476      8 6452468   1% /run
none            6452476      0 6452476   0% /run/lock
none            6452476      0 6452476   0% /run/shm
none            6452476      0 6452476   0% /run/user
tmpfs           6452476      0 6452476   0% /sys/fs/cgroup
drivers         499448504 123221896 376226608  25% /usr/lib/wsl/drivers
lib             499448504 123221896 376226608  25% /usr/lib/wsl/lib
C:\             499448504 123221896 376226608  25% /mnt/c
fische11@DESKTOP-5MHG39V:~/test$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sdb        251G  1.9G  237G   1% /
tmpfs           6.2G   0 6.2G   0% /mnt/wsl
tools          477G  118G  359G  25% /init
none            6.2G   0 6.2G   0% /dev
none            6.2G  8.0K  6.2G   1% /run
none            6.2G   0 6.2G   0% /run/lock
none            6.2G   0 6.2G   0% /run/shm
none            6.2G   0 6.2G   0% /run/user
tmpfs           6.2G   0 6.2G   0% /sys/fs/cgroup
drivers         477G  118G  359G  25% /usr/lib/wsl/drivers
lib             477G  118G  359G  25% /usr/lib/wsl/lib
C:\             477G  118G  359G  25% /mnt/c
fische11@DESKTOP-5MHG39V:~/test$
```

### 1.5.3.6 du

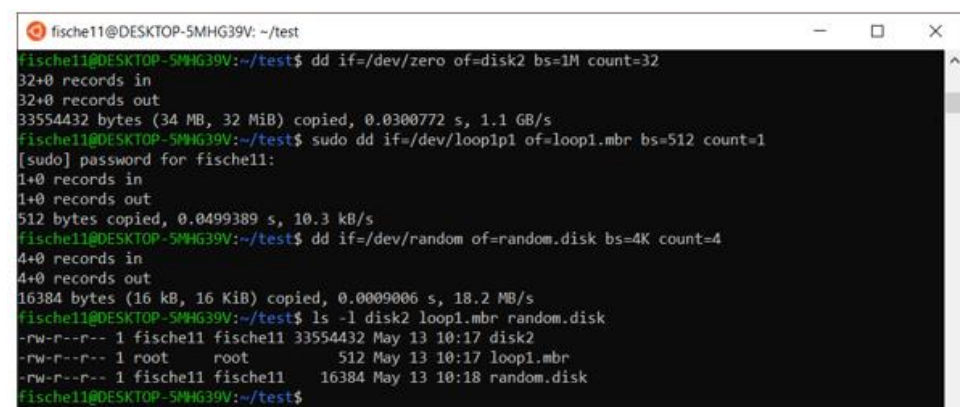
„du“ steht für disk usage. Mit diesem Befehl wird die Größe von Verzeichnissen angezeigt. Mit -d kann die maximale Tiefe angegeben werden, mit -h wird die Größe in menschenlesbarer Form ausgegeben, mit -s wird die Summe der Größen mit ausgegeben und mit --exclude werden Daten angegeben, die ignoriert werden sollen.



```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ du
4      ./backup
4      ./git/branches
56     ./git/hooks
8      ./git/objects/c3
8      ./git/objects/ad
4      ./git/objects/pack
4      ./git/objects/info
8      ./git/objects/97
36     ./git/objects
8      ./git/info
4      ./git/refs/tags
8      ./git/refs/heads
16     ./git/refs
8      ./git/logs/refs/heads
12     ./git/logs/refs
20     ./git/logs
164    ./git
71096  .
fische11@DESKTOP-5MHG39V:~/test$ du -hd 1
4.0K   ./backup
164K   ./git
70M    .
fische11@DESKTOP-5MHG39V:~/test$ du -hs ~
72M    /home/fische11
fische11@DESKTOP-5MHG39V:~/test$
```

### 1.5.3.7 dd

Der Befehl „dd“ steht für duplicate data und wird für das Klonen von Festplatten und das Erstellen großer Dateien genutzt. Um eine Festplatte vollständig zu klonen verwendet man „dd if=<Festplatte> of=<Datei>“. Weitere Optionen sind möglich: „dd if=<Eingabe> of=<Ausgabe> bs=<Stückgröße> count=<Anzahl> seek=<Ausgabefüllung> skip=<Eingabe überspringen>“.

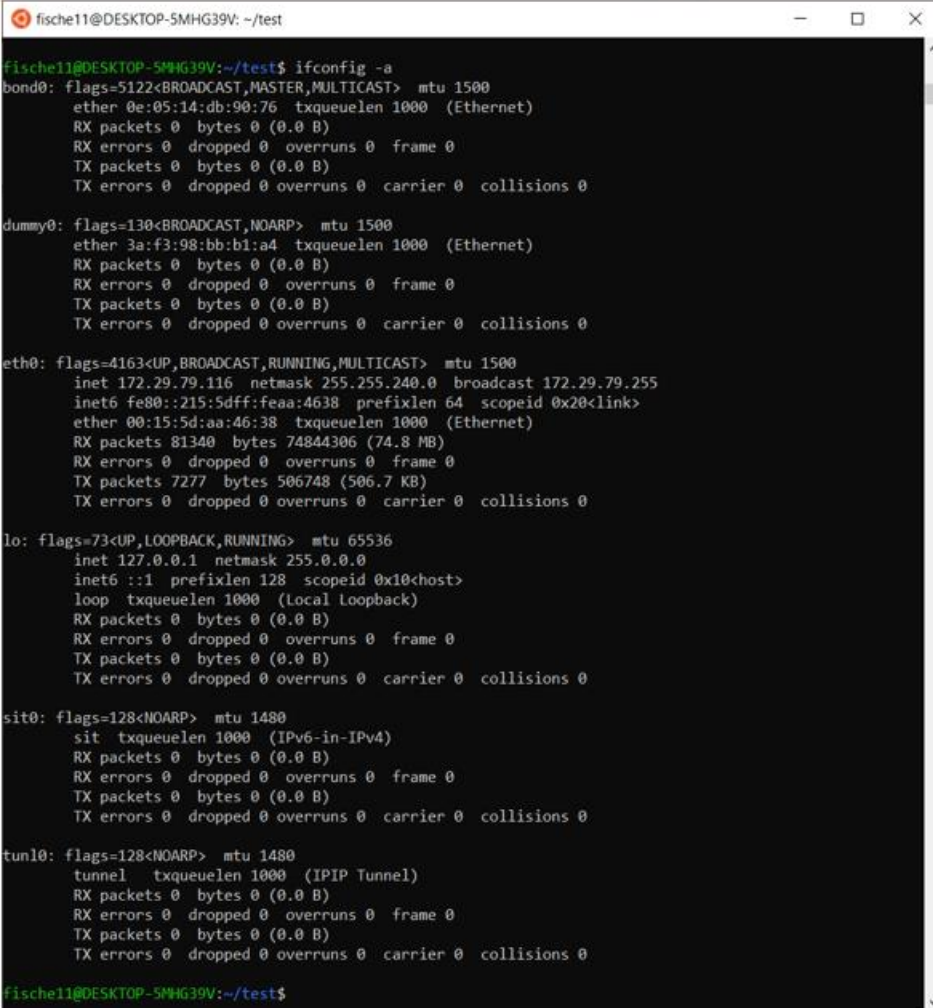


```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ dd if=/dev/zero of=disk2 bs=1M count=32
32+0 records in
32+0 records out
33554432 bytes (34 MB, 32 MiB) copied, 0.0300772 s, 1.1 GB/s
fische11@DESKTOP-5MHG39V:~/test$ sudo dd if=/dev/loop1p1 of=loop1.mbr bs=512 count=1
[sudo] password for fische11:
1+0 records in
1+0 records out
512 bytes copied, 0.0499389 s, 10.3 kB/s
fische11@DESKTOP-5MHG39V:~/test$ dd if=/dev/random of=random.disk bs=4K count=4
4+0 records in
4+0 records out
16384 bytes (16 kB, 16 KiB) copied, 0.0009006 s, 18.2 MB/s
fische11@DESKTOP-5MHG39V:~/test$ ls -l disk2 loop1.mbr random.disk
-rw-r--r-- 1 fische11 fische11 33554432 May 13 10:17 disk2
-rw-r--r-- 1 root      root          512 May 13 10:17 loop1.mbr
-rw-r--r-- 1 fische11 fische11 16384 May 13 10:18 random.disk
fische11@DESKTOP-5MHG39V:~/test$
```

## 1.5.4 Netzwerk

### 1.5.4.1 ifconfig

Mit „ifconfig“ kann die Netzwerkkonfiguration ausgegeben werden. Mit dem Parameter -a wird alles ausgegeben.



```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ ifconfig -a
bond0: flags=512<BROADCAST,MASTER,MULTICAST> mtu 1500
       ether 0e:05:14:db:90:76 txqueuelen 1000 (Ethernet)
       RX packets 0 bytes 0 (0.0 B)
       RX errors 0 dropped 0 overruns 0 frame 0
       TX packets 0 bytes 0 (0.0 B)
       TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

dummy0: flags=130<BROADCAST,NOARP> mtu 1500
       ether 3a:f3:98:bb:b1:a4 txqueuelen 1000 (Ethernet)
       RX packets 0 bytes 0 (0.0 B)
       RX errors 0 dropped 0 overruns 0 frame 0
       TX packets 0 bytes 0 (0.0 B)
       TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
       inet 172.29.79.116 netmask 255.255.240.0 broadcast 172.29.79.255
       inet6 fe80::215:5dff:feaa:4638 prefixlen 64 scopeid 0x20<link>
       ether 00:15:5d:aa:46:38 txqueuelen 1000 (Ethernet)
       RX packets 81340 bytes 74844306 (74.8 MB)
       RX errors 0 dropped 0 overruns 0 frame 0
       TX packets 7277 bytes 506748 (506.7 KB)
       TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
       inet 127.0.0.1 netmask 255.0.0.0
       inet6 ::1 prefixlen 128 scopeid 0x10<host>
       loop txqueuelen 1000 (Local Loopback)
       RX packets 0 bytes 0 (0.0 B)
       RX errors 0 dropped 0 overruns 0 frame 0
       TX packets 0 bytes 0 (0.0 B)
       TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

sit0: flags=128<NOARP> mtu 1480
       sit txqueuelen 1000 (IPv6-in-IPv4)
       RX packets 0 bytes 0 (0.0 B)
       RX errors 0 dropped 0 overruns 0 frame 0
       TX packets 0 bytes 0 (0.0 B)
       TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

tunl0: flags=128<NOARP> mtu 1480
       tunnel txqueuelen 1000 (IPIP Tunnel)
       RX packets 0 bytes 0 (0.0 B)
       RX errors 0 dropped 0 overruns 0 frame 0
       TX packets 0 bytes 0 (0.0 B)
       TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

fische11@DESKTOP-5MHG39V:~/test$
```

### 1.5.4.2 ip

Ziel von „ip“ ist das Ersetzen von if-tools wie „ifconfig“, „ifup“, „ifdown“, etc. Mit „ip“ kann die Netzwerkkonfiguration ausgelesen und modifiziert werden. Weiterhin erhält man Informationen zu MAC-Adressen, IP-Adressen und Routen.

```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: bond0: <BROADCAST,MULTICAST,MASTER> mtu 1500 qdisc noop state DOWN group default qlen 1000
   link/ether 0e:05:14:db:90:76 brd ff:ff:ff:ff:ff:ff
3: dummy0: <BROADCAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 1000
   link/ether 3a:f3:98:bb:b1:a4 brd ff:ff:ff:ff:ff:ff
4: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
   link/ether 00:15:5d:aa:46:38 brd ff:ff:ff:ff:ff:ff
   inet 172.29.79.116/20 brd 172.29.79.255 scope global eth0
       valid_lft forever preferred_lft forever
   inet6 fe80::215:5dff:feaa:4638/64 scope link
       valid_lft forever preferred_lft forever
5: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
   link/ipip 0.0.0.0 brd 0.0.0.0
6: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
   link/sit 0.0.0.0 brd 0.0.0.0
fische11@DESKTOP-5MHG39V:~/test$ ip -br addr
lo                UNKNOWN        127.0.0.1/8 ::1/128
bond0             DOWN
dummy0           DOWN
eth0             UP              172.29.79.116/20 fe80::215:5dff:feaa:4638/64
tunl0@NONE       DOWN
sit0@NONE        DOWN
fische11@DESKTOP-5MHG39V:~/test$ ip route
default via 172.29.64.1 dev eth0
172.29.64.0/20 dev eth0 proto kernel scope link src 172.29.79.116
fische11@DESKTOP-5MHG39V:~/test$
```

### 1.5.4.3 ping

Mit „ping“ kann man die Verbindung zum Host prüfen. Mit -c kann die Anzahl an versuchen angegeben werden.

```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data:
64 bytes from 8.8.8.8: icmp_seq=1 ttl=56 time=21.8 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=56 time=20.3 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=56 time=20.5 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=56 time=26.1 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=56 time=27.0 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=56 time=34.6 ms
^C
--- 8.8.8.8 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 20.340/25.062/34.607/4.992 ms
fische11@DESKTOP-5MHG39V:~/test$ ping www.hs-mittweida.de
PING www.hs-mittweida.de (141.55.192.190) 56(84) bytes of data:
64 bytes from www.hsmw.de (141.55.192.190): icmp_seq=1 ttl=61 time=52.8 ms
64 bytes from www.hsmw.de (141.55.192.190): icmp_seq=2 ttl=61 time=39.6 ms
64 bytes from www.hsmw.de (141.55.192.190): icmp_seq=3 ttl=61 time=42.9 ms
64 bytes from www.hsmw.de (141.55.192.190): icmp_seq=4 ttl=61 time=42.2 ms
^C
--- www.hs-mittweida.de ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3115ms
rtt min/avg/max/mdev = 39.640/44.404/52.842/5.021 ms
fische11@DESKTOP-5MHG39V:~/test$
```



#### 1.5.4.4 route

Der Befehl „route“ kann die Routingtabelle manipulieren und anzeigen. Er ist als IP-Firewall nutzbar.

```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default DESKTOP-5MHG39V 0.0.0.0 UG 0 0 0 eth0
172.29.64.0 0.0.0.0 255.255.240.0 U 0 0 0 eth0
fische11@DESKTOP-5MHG39V:~/test$ ping -c 1 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data:
64 bytes from 8.8.8.8: icmp_seq=1 ttl=56 time=19.4 ms

--- 8.8.8.8 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 19.404/19.404/19.404/0.000 ms
fische11@DESKTOP-5MHG39V:~/test$ sudo route add -host 8.8.8.8 metric 10 dev lo
fische11@DESKTOP-5MHG39V:~/test$ route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default DESKTOP-5MHG39V 0.0.0.0 UG 0 0 0 eth0
dns.google 0.0.0.0 255.255.255.255 UH 10 0 0 lo
172.29.64.0 0.0.0.0 255.255.240.0 U 0 0 0 eth0
fische11@DESKTOP-5MHG39V:~/test$ ping -c 1 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data:

--- 8.8.8.8 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms

fische11@DESKTOP-5MHG39V:~/test$ sudo route del 8.8.8.8
fische11@DESKTOP-5MHG39V:~/test$ route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default DESKTOP-5MHG39V 0.0.0.0 UG 0 0 0 eth0
172.29.64.0 0.0.0.0 255.255.240.0 U 0 0 0 eth0
fische11@DESKTOP-5MHG39V:~/test$ ping -c 1 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data:
64 bytes from 8.8.8.8: icmp_seq=1 ttl=56 time=24.7 ms

--- 8.8.8.8 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 24.714/24.714/24.714/0.000 ms
fische11@DESKTOP-5MHG39V:~/test$
```

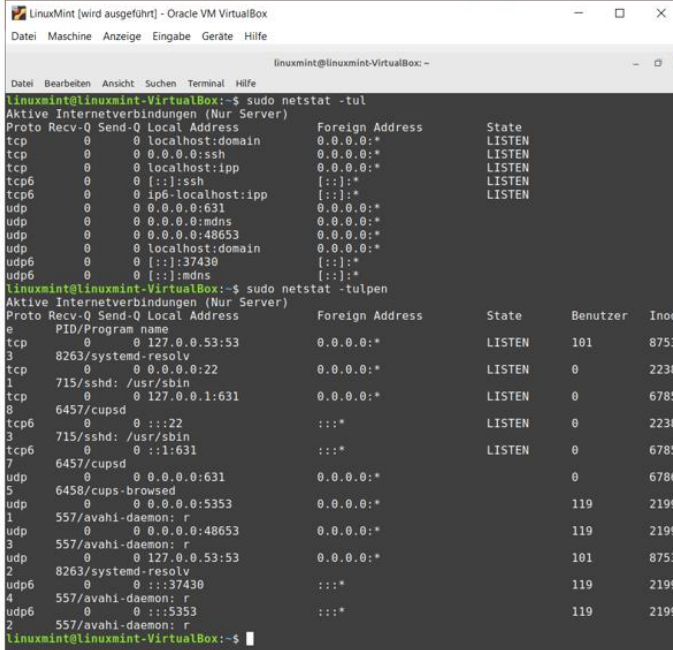
#### 1.5.4.5 arp

Mit dem Befehl „arp“ kann die ARP-Tabelle angezeigt werden. ARP steht für Address Resolution Protocol und ist für die Übersetzung zwischen MAC- und IP-Adresse zuständig.

```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ arp
Address HWtype HWaddress Flags Mask Iface
DESKTOP-5MHG39V ether 00:15:5d:2b:d7:26 C
fische11@DESKTOP-5MHG39V:~/test$ arp -a
DESKTOP-5MHG39V (172.29.64.1) at 00:15:5d:2b:d7:26 [ether] on eth0
fische11@DESKTOP-5MHG39V:~/test$
```

### 1.5.4.6 netstat

„netstat“ steht für network statistics und liefert umfassende Netzwerkinformationen. Mit -r werden Routen, mit -t TCP-Ports, mit -u UDP-Ports, mit -l offene lauschende Ports, mit -p die zugehörige Prozess-ID, mit -e erweiterte Informationen und mit -n keine DNS-Auflösung ausgegeben.



```
LinuxMint [wird ausgeführt] - Oracle VM VirtualBox
Datei Maschine Anzeige Eingabe Geräte Hilfe

linuxmint@linuxmint-VirtualBox: ~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe

linuxmint@linuxmint-VirtualBox:~$ sudo netstat -tul
Aktive Internetverbindungen (Nur Server)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:domain      0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:ssh           0.0.0.0:*               LISTEN
tcp        0      0 localhost:ipp         0.0.0.0:*               LISTEN
tcp6       0      0 [::]:ssh              [::]:*                  LISTEN
tcp6       0      0 ip6-localhost:ipp    [::]:*                  LISTEN
udp        0      0 0.0.0.0:631           0.0.0.0:*               LISTEN
udp        0      0 0.0.0.0:mdns          0.0.0.0:*               LISTEN
udp        0      0 0.0.0.0:48653         0.0.0.0:*               LISTEN
udp        0      0 localhost:domain     0.0.0.0:*               LISTEN
udp6       0      0 [::]:37430            [::]:*                  LISTEN
udp6       0      0 [::]:mdns              [::]:*                  LISTEN

linuxmint@linuxmint-VirtualBox:~$ sudo netstat -tulpen
Aktive Internetverbindungen (Nur Server)
Proto Recv-Q Send-Q Local Address           Foreign Address         State  Benutzer  Inod
e      PID/Program name
tcp        0      0 127.0.0.53:53          0.0.0.0:*               LISTEN  101        8753
3      8263/systemd-resolv
tcp        0      0 0.0.0.0:22            0.0.0.0:*               LISTEN  0          2238
1      715/sshd: /usr/sbin
tcp        0      0 127.0.0.1:631          0.0.0.0:*               LISTEN  0          6785
6      6457/cupsd
tcp6       0      0 :::22                  :::*                     LISTEN  0          2238
3      715/sshd: /usr/sbin
tcp6       0      0 :::1:631                :::*                       LISTEN  0          6785
7      6457/cupsd
udp        0      0 0.0.0.0:631           0.0.0.0:*               LISTEN  0          6786
5      6458/cups-browsed
udp        0      0 0.0.0.0:5353          0.0.0.0:*               LISTEN  119        2199
1      557/avahi-daemon: r
udp        0      0 0.0.0.0:48653         0.0.0.0:*               LISTEN  119        2199
3      557/avahi-daemon: r
udp        0      0 127.0.0.53:53          0.0.0.0:*               LISTEN  101        8753
2      8263/systemd-resolv
udp6       0      0 [::]:37430            [::]:*                  LISTEN  119        2199
4      557/avahi-daemon: r
udp6       0      0 [::]:5353              [::]:*                  LISTEN  119        2199
2      557/avahi-daemon: r

linuxmint@linuxmint-VirtualBox:~$
```

## 1.5.5 Skripte

### 1.5.5.1 Shebang

Shebang wird durch #! dargestellt und bedeutet „Dieses Skript mit dem folgenden Programm ausführen/interpretieren“. Es steht am Dokumentanfang. Für Bash-Skripte nutzt man #!/bin/bash. Alternativen für andere Skripte wären #!/bin/sh oder #!/bin/python3.

```
fische11@DESKTOP-5MHG39V: -
GNU nano 4.8 basic_info.sh
#!/bin/bash
# author: Felix Fischer
# version: 28.04.2022
#
# This script saves basic system information in the folder main_infos.
# In the case the folder exists, no information is saved
# and the skript exits with the error 1.

if [ -d main_infos ] # if folder main_info exists
then # include commands after this only when if [ ... ] was true
    # Give the user an info and exits with error 1
    echo "Folder main_infos allready exists." >&2 # write message to error output
    echo "Exit without changes." >&2 # write message to error output
    exit 1 # exit with error code 1
fi # end of if command

# continues here when folder main_infos not exists
# Make directory main_infos and enter it
mkdir main_infos
cd main_infos

# Save some system informations
uname -a > os_info # general informations about the os
cp /etc/passwd passwd # informations about users on this system
cp /etc/group group # informations about groups on this system
cat /proc/cpuinfo > cpu_info # cpu informations
df -h > storage_info # filesystem size, usage, ...

Get Help Write Out Where Is Read 29 lines Cut Text Justify Cur Pos Undo
Exit Read File Replace Paste Text To Spell Go To Line Redo
```

### 1.5.5.2 exit

„exit“ ist für das Beenden von Scripten zuständig bzw. für das Schließen einer Shell, wenn man sich nicht in einem Script befindet. „exit“ kann ebenfalls durch den Tastenkürzel Strg+d ausgeführt werden. Standardmäßig endet „exit“ erfolgreich, sprich mit dem Code 0. Ansonsten kann auch ein Error-Code übergeben werden. Das Tastenkürzel strg+c bricht eine Skript- oder Befehlsausführung ab.

```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ exit 1
```

### 1.5.5.3 \$variable

Eine Variable wird mit „Variable=Wert“ definiert. Der Zugriff erfolgt mit \$Variable. Hierbei ist zu beachten, dass weder vor noch nach dem Gleichzeichen ein Leerzeichen kommt.



```
fische11@DESKTOP-5MHG39V: ~/Uebung_Hash
fische11@DESKTOP-5MHG39V:~/Uebung_Hash$ zahl=15
fische11@DESKTOP-5MHG39V:~/Uebung_Hash$ echo $zahl
15
fische11@DESKTOP-5MHG39V:~/Uebung_Hash$ Zeichenkette=Linux
fische11@DESKTOP-5MHG39V:~/Uebung_Hash$ echo $Zeichenkette
Linux
fische11@DESKTOP-5MHG39V:~/Uebung_Hash$ dateien='ls'
fische11@DESKTOP-5MHG39V:~/Uebung_Hash$ echo $dateien
Bilder gesucht1.jpg gesucht2.jpg gesucht3.jpg gesucht4.jpg sha_sum.txt
fische11@DESKTOP-5MHG39V:~/Uebung_Hash$
```

#### 1.5.5.4 \$?

Das Fragezeichen ist eine Spezialvariable und enthält den Exit-Code des letzten Programms. Hat \$? Den Wert Null, so wurde das Programm erfolgreich beendet. Jeder andere Wert stellt einen Error-Code dar, wobei das Programm die Error-Code Bedeutung definiert. Somit gibt es keine einheitliche Festlegung der Error-Codes, abgesehen von Null.

```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ echo "Hallo"
Hallo
fische11@DESKTOP-5MHG39V:~/test$ echo $?
0
fische11@DESKTOP-5MHG39V:~/test$ passwd
Changing password for fische11.
Current password:
passwd: Authentication token manipulation error
passwd: password unchanged
fische11@DESKTOP-5MHG39V:~/test$ echo $?
10
fische11@DESKTOP-5MHG39V:~/test$
```

#### 1.5.5.5 \$#

# ist ebenfalls eine Spezialvariable und enthält die Anzahl an Übergabeparametern.

```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ cat parameter.sh
#!/bin/bash

echo $#

fische11@DESKTOP-5MHG39V:~/test$ ./parameter.sh
0
fische11@DESKTOP-5MHG39V:~/test$ ./parameter.sh eins
1
fische11@DESKTOP-5MHG39V:~/test$ ./parameter.sh eins 1
2
fische11@DESKTOP-5MHG39V:~/test$ ./parameter.sh eins 1 -b g 8 if=/dev/null
6
fische11@DESKTOP-5MHG39V:~/test$
```

### 1.5.5.6 \$1, \$2, \$3

Hierbei handelt es sich ebenfalls um Spezialvariablen zum Speichern von Parametern. \$1 enthält den ersten Parameter, \$2 den Zweiten, \$3 den Dritten, usw.

```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ cat parameter.sh
#!/bin/bash

echo 1. Parameter: $1
echo 2. Parameter: $2
echo 3. Parameter: $3
fische11@DESKTOP-5MHG39V:~/test$ ./parameter.sh
1. Parameter:
2. Parameter:
3. Parameter:
fische11@DESKTOP-5MHG39V:~/test$ ./parameter.sh eins
1. Parameter: eins
2. Parameter:
3. Parameter:
fische11@DESKTOP-5MHG39V:~/test$ ./parameter.sh eins 1
1. Parameter: eins
2. Parameter: 1
3. Parameter:
fische11@DESKTOP-5MHG39V:~/test$ ./parameter.sh eins 1 -b g 8 if=/dev/null
1. Parameter: eins
2. Parameter: 1
3. Parameter: -b
fische11@DESKTOP-5MHG39V:~/test$
```

### 1.5.5.7 test/[ ]

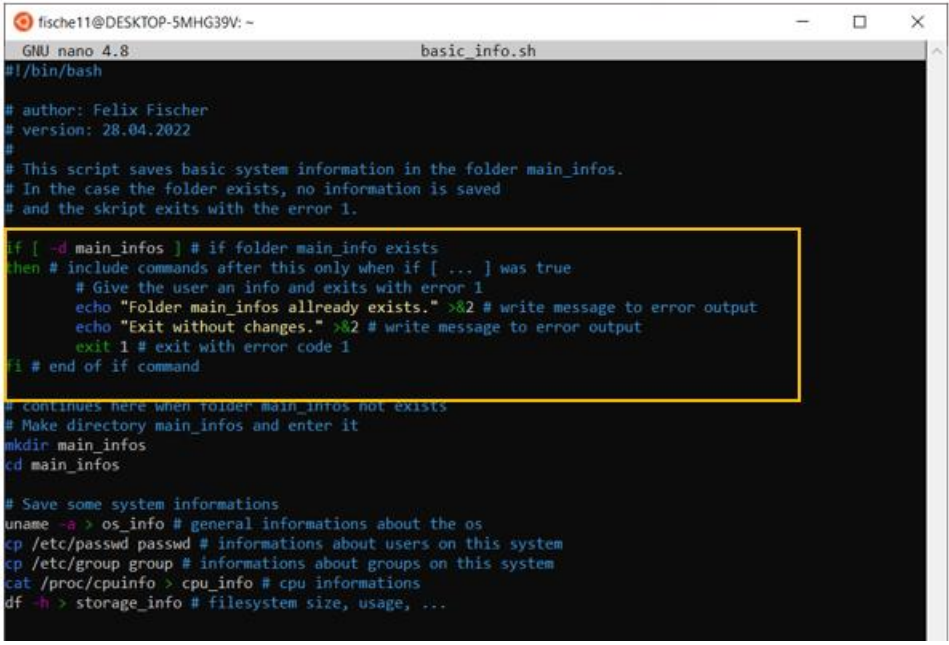
„test“ kann für die Wahrheitsprüfung verwendet werden. Die Kurzschreibweise für diese Prüfung wäre [ Aussage ]. Wird eine Null zurückgegeben, ist die Aussage wahr, bei jedem anderen zurückgegebenen Wert ist sie falsch. Die Ausgabe kann über Parameter erweitert werden. Mit -d wird geschaut, ob der Ordner, mit -f ob die reguläre Datei und mit -b ob das Blockdevice existiert. Mit -eq kann geschaut werden, ob Aussagen gleich sind, mit -lt ob eine kleiner ist, mit -gt ob eine größer ist und mit ! erfolgt die Negierung der Aussage.

```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ test -d backup
fische11@DESKTOP-5MHG39V:~/test$ echo $?
0
fische11@DESKTOP-5MHG39V:~/test$ test -d backup2
fische11@DESKTOP-5MHG39V:~/test$ echo $?
1
fische11@DESKTOP-5MHG39V:~/test$ test -f /dev/sda
fische11@DESKTOP-5MHG39V:~/test$ echo $?
1
fische11@DESKTOP-5MHG39V:~/test$ [ -b /dev/sda ]
fische11@DESKTOP-5MHG39V:~/test$ echo $?
0
fische11@DESKTOP-5MHG39V:~/test$ [ 100 -lt 101 ]
fische11@DESKTOP-5MHG39V:~/test$ echo $?
0
fische11@DESKTOP-5MHG39V:~/test$ [ 100 -eq 101 ]
fische11@DESKTOP-5MHG39V:~/test$ echo $?
1
fische11@DESKTOP-5MHG39V:~/test$ [ ! 100 -eq 101 ]
fische11@DESKTOP-5MHG39V:~/test$ echo $?
0
fische11@DESKTOP-5MHG39V:~/test$
```

### 1.5.5.8 if

Mit „if“ erfolgt eine Auswertung von test, wobei es sich hier um eine Verzweigung handelt, welche auf einer Bedingung basiert. Der Aufbau sieht wie folgt aus:

```
if [ Bedingung ]
then
    # Wenn Bedingung erfüllt
elif [ weitere Bedingung ]
    # Wenn zweite Bedingung erfüllt
else
    # Wenn keine Bedingung erfüllt
fi
```



```
fische11@DESKTOP-5MHG39V: ~
GNU nano 4.8 basic_info.sh
#!/bin/bash

# author: Felix Fischer
# version: 28.04.2022
#
# This script saves basic system information in the folder main_infos.
# In the case the folder exists, no information is saved
# and the skript exits with the error 1.

if [ -d main_infos ] # if folder main_info exists
then # include commands after this only when if [ ... ] was true
    # Give the user an info and exits with error 1
    echo "Folder main_infos already exists." >&2 # write message to error output
    echo "Exit without changes." >&2 # write message to error output
    exit 1 # exit with error code 1
fi # end of if command

# continues here when folder main_infos not exists
# Make directory main_infos and enter it
mkdir main_infos
cd main_infos

# Save some system informations
uname -a > os_info # general informations about the os
cp /etc/passwd passwd # informations about users on this system
cp /etc/group group # informations about groups on this system
cat /proc/cpuinfo > cpu_info # cpu informations
df -h > storage_info # filesystem size, usage, ...
```

### 1.5.5.9 `Subbefehl`

Die Angabe von `Subbefehl` führt dazu, dass der Befehl zuerst ausgeführt wird. Das Ergebnis ersetzt anschließend den Subbefehl.

```
fische11@DESKTOP-5MHG39V: ~/Uebung_Hash
fische11@DESKTOP-5MHG39V:~/Uebung_Hash$ sha512sum Bilder/* | grep `sha512sum gesucht1.jpg | cut -d \ -f 1`
32fa394732e00d8fe217b45581dfc0034926877fb62f8ec35166aab83ff185d11f5b15947d8268aaca109091a5dd61c22304914c
80013101fe2a36aa0b399c41 Bilder/27.jpg
fische11@DESKTOP-5MHG39V:~/Uebung_Hash$ # wird nach Subbefehlinterpertation zu:
fische11@DESKTOP-5MHG39V:~/Uebung_Hash$ sha512sum Bilder/* | grep 32fa394732e00d8fe217b45581dfc003492687
7fb62f8ec35166aab83ff185d11f5b15947d8268aaca109091a5dd61c22304914c80013101fe2a36aa0b399c41
32fa394732e00d8fe217b45581dfc0034926877fb62f8ec35166aab83ff185d11f5b15947d8268aaca109091a5dd61c22304914c
80013101fe2a36aa0b399c41 Bilder/27.jpg
fische11@DESKTOP-5MHG39V:~/Uebung_Hash$ # , da `sha512sum gesucht1.jpg | cut -d \ -f 1` folgendes ausgi
bt:
fische11@DESKTOP-5MHG39V:~/Uebung_Hash$ sha512sum gesucht1.jpg | cut -d \ -f 1
32fa394732e00d8fe217b45581dfc0034926877fb62f8ec35166aab83ff185d11f5b15947d8268aaca109091a5dd61c22304914c
80013101fe2a36aa0b399c41
fische11@DESKTOP-5MHG39V:~/Uebung_Hash$
```

### 1.5.5.10 while

Mit „while“ wird ein Abschnitt wiederholt, solange eine Bedingung gilt. Der Aufbau ist folgender:

```
while [ Bedingung ]
do
    # Anweisungen
done
```

```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ while [ -d backup ]; do echo backup existiert; sleep 1; done
backup existiert
backup existiert
backup existiert
backup existiert
backup existiert
backup existiert
backup existiert
backup existiert
backup existiert
backup existiert
backup existiert
backup existiert
backup existiert
backup existiert
backup existiert
backup existiert
backup existiert
backup existiert
backup existiert
backup existiert
fische11@DESKTOP-5MHG39V:~/test$ # Ordner backup wurde vom zweiten Terminal aus gelöscht.
fische11@DESKTOP-5MHG39V:~/test$
```

### 1.5.5.11 for

Bei „for“ handelt es sich um eine gezählte Wiederholung. Die Anzahl an Wiederholungen wird fest vorgegeben und die Wiederholung erfolgt für jedes X.

```
fische11@DESKTOP-SMHG39V: ~/test
fische11@DESKTOP-SMHG39V:~/test$ for i in 1 100 55; do echo $i; done
1
100
55
fische11@DESKTOP-SMHG39V:~/test$ for i in {1..4}; do echo $i; done
1
2
3
4
fische11@DESKTOP-SMHG39V:~/test$ for (( i=1; $i < 10; ((i=$i+3)) )); do echo $i; done
1
4
7
fische11@DESKTOP-SMHG39V:~/test$ for i in `ls /dev/sd*`; do echo $i; done
/dev/sda
/dev/sdb
fische11@DESKTOP-SMHG39V:~/test$ for i in $(seq 2 5); do echo $i; done
2
3
4
5
fische11@DESKTOP-SMHG39V:~/test$ for i in $(ls *.mbr); do echo $i; done
loop1.mbr
sda.mbr
sdb.mbr
fische11@DESKTOP-SMHG39V:~/test$
```

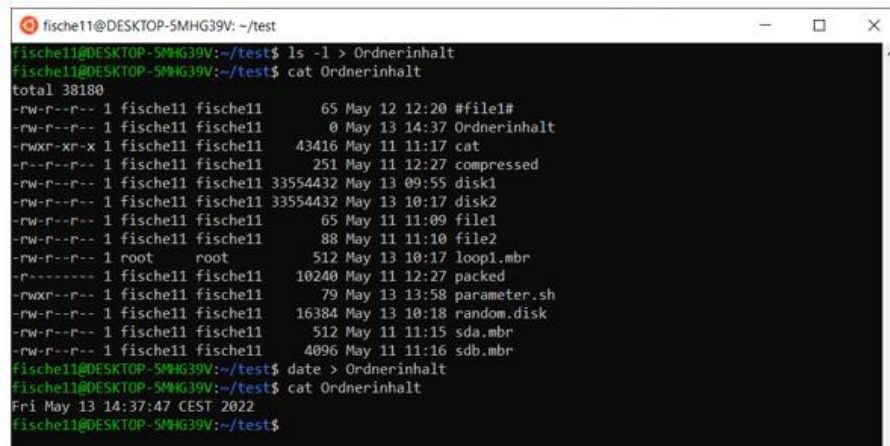
### 1.5.5.12 |

| bedeutet gesprochen „Pipe“. Hiermit erfolgt die Übergabe der Ausgabe an das nächste Programm als Eingabe. Der Aufbau ist folgender: Befehl1 | Befehl2. Das Zeichen | ist ein senkrechter Strich auf der Tastatur bei den Zeichen „<“ und „>“. Er wird mit der Tastenkombination AltGr+< erzeugt.

```
fische11@DESKTOP-SMHG39V: ~
fische11@DESKTOP-SMHG39V:~$ cat basic_info.sh | nl -b a -n 1n
1  #!/bin/bash
2
3  # author: Felix Fischer
4  # version: 28.04.2022
5  #
6  # This script saves basic system information in the folder main_infos.
7  # In the case the folder exists, no information is saved
8  # and the skript exits with the error 1.
9
10 if [ -d main_infos ] # if folder main_info exists
11 then # include commands after this only when if [ ... ] was true
12     # Give the user an info and exits with error 1
13     echo "Folder main_infos allready exists." >&2 # write message to error output
14     echo "Exit without changes." >&2 # write message to error output
15     exit 1 # exit with error code 1
16 fi # end of if command
17
18 # continues here when folder main_infos not exists
19 # Make directory main_infos and enter it
20 mkdir main_infos
21 cd main_infos
22
23 # Save some system informations
24 uname -a > os_info # general informations about the os
25 cp /etc/passwd passwd # informations about users on this system
26 cp /etc/group group # informations about groups on this system
27 cat /proc/cpuinfo > cpu_info # cpu informations
28 df -h > storage_info # filesystem size, usage, ...
29
fische11@DESKTOP-SMHG39V:~$ cat basic_info.sh | nl -b a -n 1n | tail -n 1
29
fische11@DESKTOP-SMHG39V:~$ cat basic_info.sh | nl -b a -n 1n | tail -n 1 | cut -d \ -f 1
29
fische11@DESKTOP-SMHG39V:~$ ls -l test | grep -v fische11
total 38184
-rw-r--r-- 1 root root 512 May 13 10:17 loop1.mbr
fische11@DESKTOP-SMHG39V:~$
```

### 1.5.5.13 >

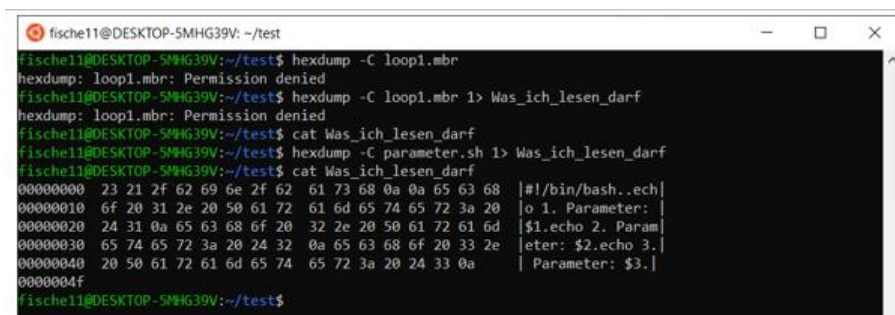
Mit „>“ kann die Ausgabe eines Befehls in eine Datei umgeleitet werden. Hierbei wird die Datei überschrieben. Der Aufbau ist folgender: Befehl > Dateiname



```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ ls -l > Ordnerinhalt
fische11@DESKTOP-5MHG39V:~/test$ cat Ordnerinhalt
total 38180
-rw-r--r-- 1 fische11 fische11    65 May 12 12:20 #file1#
-rw-r--r-- 1 fische11 fische11     0 May 13 14:37 Ordnerinhalt
-rwxr-xr-x 1 fische11 fische11  43416 May 11 11:17 cat
-r--r--r-- 1 fische11 fische11    251 May 11 12:27 compressed
-rw-r--r-- 1 fische11 fische11 33554432 May 13 09:55 disk1
-rw-r--r-- 1 fische11 fische11 33554432 May 13 10:17 disk2
-rw-r--r-- 1 fische11 fische11    65 May 11 11:09 file1
-rw-r--r-- 1 fische11 fische11    88 May 11 11:10 file2
-rw-r--r-- 1 root     root       512 May 13 10:17 loop1.mbr
-r----- 1 fische11 fische11  10240 May 11 12:27 packed
-rwxr--r-- 1 fische11 fische11    79 May 13 13:58 parameter.sh
-rw-r--r-- 1 fische11 fische11  16384 May 13 10:18 random.disk
-rw-r--r-- 1 fische11 fische11    512 May 11 11:15 sda.mbr
-rw-r--r-- 1 fische11 fische11   4096 May 11 11:16 sdb.mbr
fische11@DESKTOP-5MHG39V:~/test$ date > Ordnerinhalt
fische11@DESKTOP-5MHG39V:~/test$ cat Ordnerinhalt
Fri May 13 14:37:47 CEST 2022
fische11@DESKTOP-5MHG39V:~/test$
```

### 1.5.5.14 1>

Mit „1>“ wird die Standardausgabe in eine Datei umgeleitet, wobei die Datei überschrieben wird. Die Ausgabe erfolgt nur auf der Standardausgabe im Skript mit dem Befehl „>&1“.



```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ hexdump -C loop1.mbr
hexdump: loop1.mbr: Permission denied
fische11@DESKTOP-5MHG39V:~/test$ hexdump -C loop1.mbr 1> Was_ich_lesen_darf
hexdump: loop1.mbr: Permission denied
fische11@DESKTOP-5MHG39V:~/test$ cat Was_ich_lesen_darf
fische11@DESKTOP-5MHG39V:~/test$ hexdump -C parameter.sh 1> Was_ich_lesen_darf
fische11@DESKTOP-5MHG39V:~/test$ cat Was_ich_lesen_darf
00000000 23 21 2f 62 69 6e 2f 62 61 73 68 0a 0a 65 63 68 |#/bin/bash..ech|
00000010 6f 20 31 2e 20 50 61 72 61 6d 65 74 65 72 3a 20 |o 1. Parameter: |
00000020 24 31 0a 65 63 68 6f 20 32 2e 20 50 61 72 61 6d |$1.echo 2. Param|
00000030 65 74 65 72 3a 20 24 32 0a 65 63 68 6f 20 33 2e |eter: $2.echo 3.|
00000040 20 50 61 72 61 6d 65 74 65 72 3a 20 24 33 0a | Parameter: $3.|
0000004f
fische11@DESKTOP-5MHG39V:~/test$
```

### 1.5.5.15 2>

„2>“ wird verwendet, um die Errorausgabe in eine Datei umzuleiten, wobei die Datei überschrieben wird. Die Ausgabe erfolgt nur auf der Errorausgabe im Skript mit dem Befehl „>&2“.



```

fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ hexdump -C loop1.mbr
hexdump: loop1.mbr: Permission denied
fische11@DESKTOP-5MHG39V:~/test$ hexdump -C loop1.mbr 2> fehler
fische11@DESKTOP-5MHG39V:~/test$ cat fehler
hexdump: loop1.mbr: Permission denied
fische11@DESKTOP-5MHG39V:~/test$ hexdump -C parameter.sh 2> fehler
00000000 23 21 2f 62 69 6e 2f 62 61 73 68 0a 0a 65 63 68 |#!/bin/bash.ech|
00000010 6f 20 31 2e 20 50 61 72 61 6d 65 74 65 72 3a 20 |o 1. Parameter: |
00000020 24 31 0a 65 63 68 6f 20 32 2e 20 50 61 72 61 6d |$1.echo 2. Param|
00000030 65 74 65 72 3a 20 24 32 0a 65 63 68 6f 20 33 2e |eten: $2.echo 3.|
00000040 20 50 61 72 61 6d 65 74 65 72 3a 20 24 33 0a | Parameter: $3.|
0000004f
fische11@DESKTOP-5MHG39V:~/test$ cat fehler
fische11@DESKTOP-5MHG39V:~/test$

```

1.5.5.16 >>

„>>“ wird verwendet, um die Ausgabe in eine Datei umzuleiten, wobei die Ausgabe an die Datei angehängen wird, anstatt sie zu überschreiben. „1>>“ ist für die Standardausgabe und „2>>“ für die Errorausgabe.

```

fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ ls -l >> Ordnerinhalt
fische11@DESKTOP-5MHG39V:~/test$ cat Ordnerinhalt
total 38184
-rw-r--r-- 1 fische11 fische11      65 May 12 12:20 #file1#
-rw-r--r-- 1 fische11 fische11       0 May 13 14:46 Ordnerinhalt
-rw-r--r-- 1 fische11 fische11    403 May 13 14:42 Was_ich_lesen_darf
-rwxr-xr-x 1 fische11 fische11   43416 May 11 11:17 cat
-r--r--r-- 1 fische11 fische11    251 May 11 12:27 compressed
-rw-r--r-- 1 fische11 fische11 33554432 May 13 09:55 disk1
-rw-r--r-- 1 fische11 fische11 33554432 May 13 10:17 disk2
-rw-r--r-- 1 fische11 fische11       0 May 13 14:44 fehler
-rw-r--r-- 1 fische11 fische11    65 May 11 11:09 file1
-rw-r--r-- 1 fische11 fische11    88 May 11 11:10 file2
-rw----- 1 root      root        512 May 13 10:17 loop1.mbr
-r----- 1 fische11 fische11   10240 May 11 12:27 packed
-rwxr--r-- 1 fische11 fische11     79 May 13 13:58 parameter.sh
-rw-r--r-- 1 fische11 fische11   16384 May 13 10:18 random.disk
-rw-r--r-- 1 fische11 fische11    512 May 11 11:15 sda.mbr
-rw-r--r-- 1 fische11 fische11   4096 May 11 11:16 sdb.mbr
fische11@DESKTOP-5MHG39V:~/test$ date >> Ordnerinhalt
fische11@DESKTOP-5MHG39V:~/test$ cat Ordnerinhalt
total 38184
-rw-r--r-- 1 fische11 fische11      65 May 12 12:20 #file1#
-rw-r--r-- 1 fische11 fische11       0 May 13 14:46 Ordnerinhalt
-rw-r--r-- 1 fische11 fische11    403 May 13 14:42 Was_ich_lesen_darf
-rwxr-xr-x 1 fische11 fische11   43416 May 11 11:17 cat
-r--r--r-- 1 fische11 fische11    251 May 11 12:27 compressed
-rw-r--r-- 1 fische11 fische11 33554432 May 13 09:55 disk1
-rw-r--r-- 1 fische11 fische11 33554432 May 13 10:17 disk2
-rw-r--r-- 1 fische11 fische11       0 May 13 14:44 fehler
-rw-r--r-- 1 fische11 fische11    65 May 11 11:09 file1
-rw-r--r-- 1 fische11 fische11    88 May 11 11:10 file2
-rw----- 1 root      root        512 May 13 10:17 loop1.mbr
-r----- 1 fische11 fische11   10240 May 11 12:27 packed
-rwxr--r-- 1 fische11 fische11     79 May 13 13:58 parameter.sh
-rw-r--r-- 1 fische11 fische11   16384 May 13 10:18 random.disk
-rw-r--r-- 1 fische11 fische11    512 May 11 11:15 sda.mbr
-rw-r--r-- 1 fische11 fische11   4096 May 11 11:16 sdb.mbr
Fri May 13 14:46:31 CEST 2022
fische11@DESKTOP-5MHG39V:~/test$

```

1.5.5.17 <

Bei „<“ dient der Dateinhalt als Input. Es kann in Kombination mit der Pipe wie folgt genutzt werden: „Befehl1 < Input | Befehl2“.

```

fische11@DESKTOP-SMHG39V: ~/test
fische11@DESKTOP-SMHG39V:~/test$ ls -l > Ordnerinhalt
fische11@DESKTOP-SMHG39V:~/test$ cat Ordnerinhalt
total 38188
-rw-r--r-- 1 fische11 fische11    65 May 12 12:20 #file1#
-rw-r--r-- 1 fische11 fische11     0 May 13 14:54 Ordnerinhalt
-rw-r--r-- 1 fische11 fische11   145 May 13 14:54 Ordnerinhalt2
-rw-r--r-- 1 fische11 fische11   403 May 13 14:42 Was_ich_lesen_darf
-rwxr-xr-x 1 fische11 fische11  43416 May 11 11:17 cat
-r--r--r-- 1 fische11 fische11   251 May 11 12:27 compressed
-rw-r--r-- 1 fische11 fische11 33554432 May 13 09:55 disk1
-rw-r--r-- 1 fische11 fische11 33554432 May 13 10:17 disk2
-rw-r--r-- 1 fische11 fische11     0 May 13 14:44 fehler
-rw-r--r-- 1 fische11 fische11    65 May 11 11:09 file1
-rw-r--r-- 1 fische11 fische11    88 May 11 11:10 file2
-rw----- 1 root      root      512 May 13 10:17 loop1.mbr
-r----- 1 fische11 fische11 10240 May 11 12:27 packed
-rwxr--r-- 1 fische11 fische11    79 May 13 13:58 parameter.sh
-rw-r--r-- 1 fische11 fische11 16384 May 13 10:18 random.disk
-rw-r--r-- 1 fische11 fische11   512 May 11 11:15 sda.mbr
-rw-r--r-- 1 fische11 fische11  4096 May 11 11:16 sdb.mbr
fische11@DESKTOP-SMHG39V:~/test$ cat < Ordnerinhalt | cut -b 54- > Dateinamen
fische11@DESKTOP-SMHG39V:~/test$ cat Dateinamen
#file1#
Ordnerinhalt
Ordnerinhalt2
Was_ich_lesen_darf
cat
compressed
disk1
disk2
fehler
file1
file2
loop1.mbr
packed
parameter.sh
random.disk
sda.mbr
sdb.mbr
fische11@DESKTOP-SMHG39V:~/test$

```

### 1.5.6 Zusammenfassung

Sie kennen nun folgende Konsoleneditoren: nano , vi, vim, emacs und hexeditor.

Außerdem haben Sie heute folgende Anwendungen kennengelernt: uname, top, ps, kill, date, lsusb, lspci, lscpu, lsmem, gdisk, mkfs, mount, umount, df, du, dd, ifconfig, ip, ping, route, arp, netstat, exit, test, if, for, while, do und done.

Von den speziellen Variablen kennen Sie: \$?, \$#, \$1, \$2, \$3, ...

Mit Variablen, Subbefehlen und Aus- und Eingabengabenumleitung in Bash sind Sie nun vertraut.

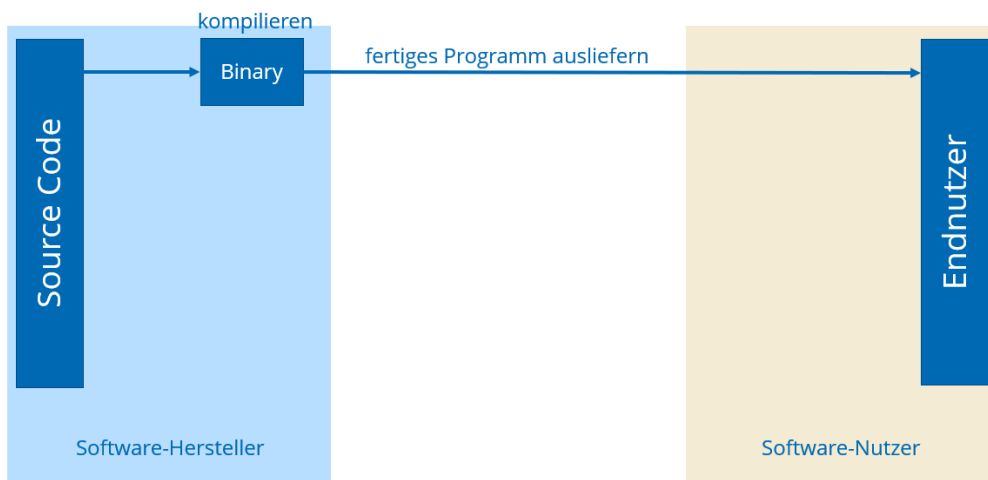
Sie wissen die verschiedenen Umleitungsvarianten der Ausgabe. |, >, >>, 1>, 2>, 1>>, 2>>, >&1, >&2 und <



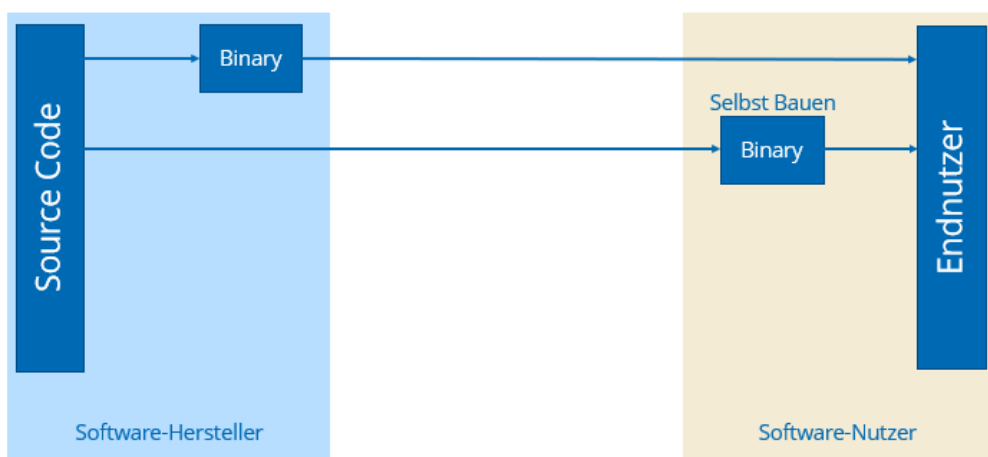
## 1.6 Linux Paket- und Dienstmanagement

### 1.6.1 Softwareinstallationswege

Es gibt verschiedene Wege, um Software zu installieren. Eine Option wäre, dass der Software-Hersteller den Source Code in Form einer Binary kompiliert und anschließend das fertige Programm an den Software-Nutzer ausliefert.

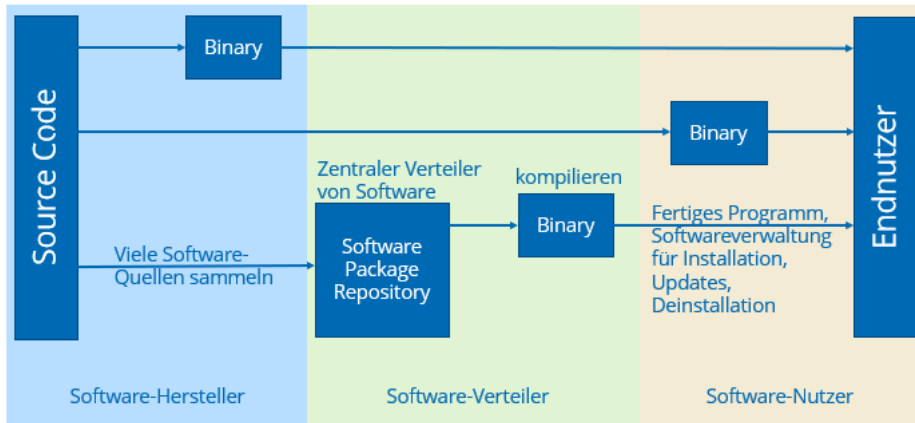


Eine weitere Option wäre, dass der Software-Hersteller dem Software-Nutzer den Source-Code zur Verfügung stellt. Der Software-Nutzer erstellt selbst eine Binary.

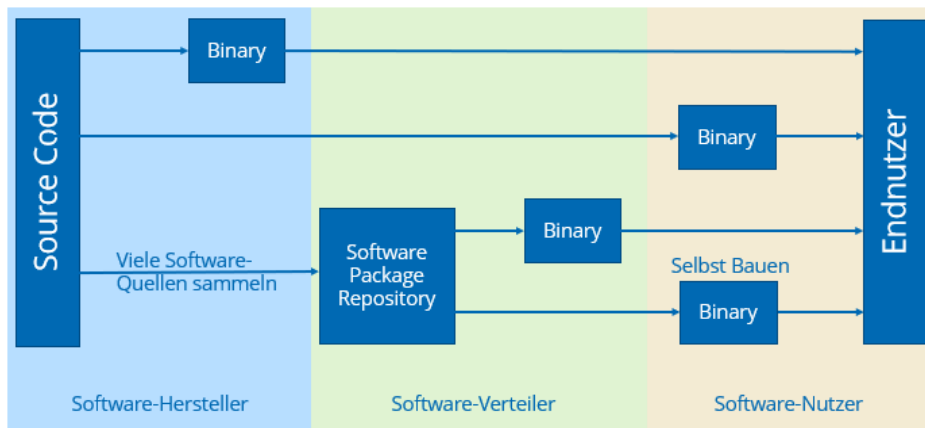


Die dritte Möglichkeit wäre, dass der Software-Hersteller Source Code aus vielen Software-Quellen sammelt. Diese werden an einen Software-Verteiler gegeben, welcher diese zu einem Software Package Repository zusammenfügt. Dieses dient als zentraler Verteiler von Software.

Aus dem Software-Package Repository werden dann Binaries erstellt und kompiliert. Diese werden als fertiges Programm oder als Softwareverwaltung für die Installation, Updates und Deinstallation an den Software-Nutzer gegeben.

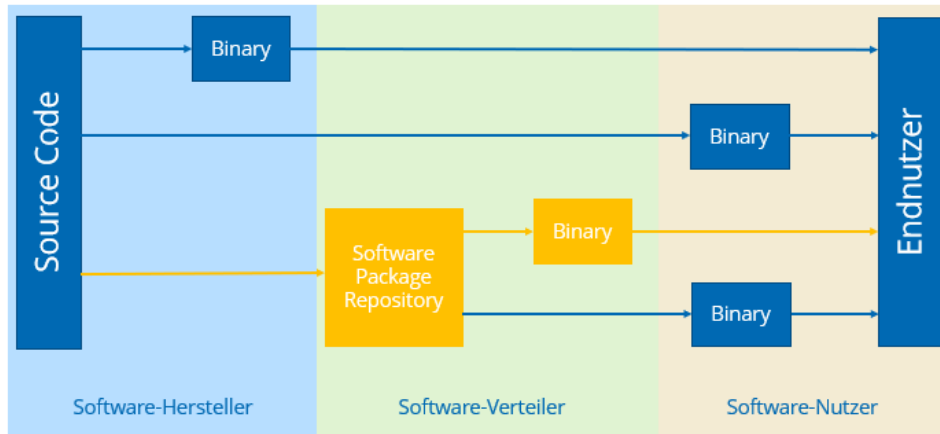


Die letzte Option wäre, dass der Software-Nutzer selbst den Code aus dem Software Package Repository nimmt und die Binary selbst baut.



## 1.6.2 Paketverwaltung

Bei der Paketverwaltung wird die dritte beschriebene Möglichkeit der Softwareinstallation verwendet:



### 1.6.2.1 Aufgaben

Bei der Paketverwaltung gibt es verschiedene Aufgaben. Bezüglich der Software ist die Paketverwaltung für das Installieren, Aktualisieren, Konfigurieren und Deinstallieren zuständig. Außerdem liefert sie Metainformationen zur Software wie die Version, den Hersteller, Abhängigkeiten oder Ähnliches. Weiterhin ist die Paketverwaltung für die Sicherheit zuständig, indem Updates durchgeführt werden. Weitere Sicherheitsfeatures sind das Signieren der Software für Wahrung der Integrität und bestimmte Programmfunktionen, die bei Deinstallation anderer Software eine Ausfallsicherheit gewährleisten.

### 1.6.2.2 Softwareorganisation

Die Software wird in Einzelkomponenten, sogenannte Pakete, zerlegt. Es können Abhängigkeiten festgelegt und eine Standardkonfiguration angeboten werden. Dies hat einige Vorteile. Mehrfach genutzte Pakete müssen nur einmalig installiert werden. Weiterhin wird weniger Speicherplatz benötigt und eine Trennung von Programm und Konfiguration wird realisiert. Insgesamt ist die Softwareverwaltung dadurch einfach gehalten.

### 1.6.2.3 Paketformate

Es gibt verschiedene Paketformate, dazu zählen:

- deb (Debian Build)
  - Debian, ubuntu, Linux Mint, ...
- dpkg (Debian Package, Vereinfachung für Endnutzer von deb)
  - Debian, ubuntu, Linux Mint, ...
- rpm (Redhat Package Manager)

- Redhat, Fedora, CentOS, openSuse, Solaris, ...
- ebuild, Portage
  - Gentoo
- PiSi (Packages Installed Successfully as Intended)
  - Pardus, Solus
- pub und pet
  - Puppy
- iPKG
  - OpenVRT

#### 1.6.2.4 Linux Package Manager

Unter Linux gibt es verschiedene Package Manager. Dazu gehören:

- apt-get
  - Debian, ubuntu, Linux Mint, ...
- apt
  - Debian, ubuntu, Linux Mint, ...
- pacman
  - Arch Linux
- yum
  - Redhat, Fedora
- YaST
  - OpenSUSE
- emerge und Portage
  - Gentoo
- PiSi
  - Pardus, Solus

#### 1.6.2.5 Package Manager für Programmierer

Für Programmierer gibt es zusätzliche Package Manager, je nach dem mit welcher Programmiersprache gearbeitet wird:

- pip3
  - Python Package Manager
- npm
  - Node Package Manager, JavaScript
- NuGet
  - C# und Visual Studio
- Perl Package Manager
  - Perl

### 1.6.2.6 Paketverwaltung am Beispiel von apt

#### Paketinformationen aktualisieren

Paketinformationen können mit dem Befehl „apt update“ aktualisiert werden. Die Quellen für die Aktualisierung sind unter „/etc/apt/sources.list“ und unter „/etc/apt/sources.list.d/\*“ definiert. Man kann sehen welche Pakete es gibt, welche Version aktuell ist, weiterhin noch Metainformationen einsehen, man kann Unterschiede auflösen und Updates installieren.

```
fische11@DESKTOP-5MHG39V: ~  
fische11@DESKTOP-5MHG39V:~$ sudo apt update  
[sudo] password for fische11:  
Hit:1 http://archive.ubuntu.com/ubuntu focal InRelease  
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]  
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]  
Get:4 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [1450 kB]  
Get:5 http://archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]  
Get:6 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1790 kB]  
Get:7 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [249 kB]  
Get:8 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [10.2 kB]  
Get:9 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [911 kB]  
Get:10 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en [130 kB]  
Get:11 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [702 kB]  
Get:12 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [125 kB]  
Get:13 http://security.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [14.4 kB]  
Get:14 http://archive.ubuntu.com/ubuntu focal-updates/main Translation-en [330 kB]  
Get:15 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [15.1 kB]  
Get:16 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [975 kB]  
Get:17 http://archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [139 kB]  
Get:18 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [920 kB]  
Get:19 http://archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [206 kB]  
Get:20 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [20.6 kB]  
Get:21 http://archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [42.2 kB]  
Get:22 http://archive.ubuntu.com/ubuntu focal-backports/main Translation-en [10.1 kB]  
Get:23 http://archive.ubuntu.com/ubuntu focal-backports/main amd64 c-n-f Metadata [864 B]  
Get:24 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [22.7 kB]  
Get:25 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 c-n-f Metadata [804 B]  
Fetched 8482 kB in 10s (816 kB/s)  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
32 packages can be upgraded. Run 'apt list --upgradable' to see them.  
fische11@DESKTOP-5MHG39V:~$
```

#### Programme installieren

Mit „apt install <Programm>“ können Programme installiert werden. Hierbei werden benötigte Pakete aufgelöst. Zusätzlich wird die Downloadgröße und die Installationsgröße bestimmt. Weiterhin werden hiermit Pakete heruntergeladen und installiert.

```
fische11@DESKTOP-SMHG39V: ~  
fische11@DESKTOP-SMHG39V:~$ sudo apt install gdb  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following package was automatically installed and is no longer required:  
  libfwdppplugin1  
Use 'sudo apt autoremove' to remove it.  
The following additional packages will be installed:  
  gdbserver libbabeltrace1 libc6 libc6-dbg libcc1-0 libdw1  
Suggested packages:  
  gdb-doc glibc-doc  
The following NEW packages will be installed:  
  gdb gdbserver libbabeltrace1 libc6-dbg libcc1-0 libdw1  
The following packages will be upgraded:  
  libc6  
1 upgraded, 6 newly installed, 0 to remove and 31 not upgraded.  
Need to get 26.6 MB of archives.  
After this operation, 42.3 MB of additional disk space will be used.  
Do you want to continue? [Y/n]  
Get:1 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libc6 amd64 2.31-0ubuntu9.9 [2722 kB]  
Get:2 http://archive.ubuntu.com/ubuntu focal/main amd64 libdw1 amd64 0.176-1.1build1 [226 kB]  
Get:3 http://archive.ubuntu.com/ubuntu focal/main amd64 libbabeltrace1 amd64 1.5.8-1build1 [156 kB]  
Get:4 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 gdb amd64 9.2-0ubuntu1-20.04.1 [3222 kB]  
Get:5 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 gdbserver amd64 9.2-0ubuntu1-20.04.1 [22  
2 kB]  
Get:6 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libcc1-0 amd64 10.3.0-1ubuntu1-20.04 [48  
.8 kB]  
Get:7 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libc6-dbg amd64 2.31-0ubuntu9.9 [20.0 MB  
]  
Fetched 26.6 MB in 15s (1792 kB/s)  
Preconfiguring packages ...  
(Reading database ... 53155 files and directories currently installed.)  
Preparing to unpack .../libc6_2.31-0ubuntu9.9_amd64.deb ...  
Unpacking libc6:amd64 (2.31-0ubuntu9.9) over (2.31-0ubuntu9.7) ...  
Setting up libc6:amd64 (2.31-0ubuntu9.9) ...  
Selecting previously unselected package libdw1:amd64.  
(Reading database ... 53155 files and directories currently installed.)  
Preparing to unpack .../0-libdw1_0.176-1.1build1_amd64.deb ...  
Unpacking libdw1:amd64 (0.176-1.1build1) ...  
Selecting previously unselected package libbabeltrace1:amd64.  
Preparing to unpack .../1-libbabeltrace1_1.5.8-1build1_amd64.deb ...  
Unpacking libbabeltrace1:amd64 (1.5.8-1build1) ...  
Selecting previously unselected package gdb.  
Preparing to unpack .../2-gdb_9.2-0ubuntu1-20.04.1_amd64.deb ...  
Unpacking gdb (9.2-0ubuntu1-20.04.1) ...  
Selecting previously unselected package gdbserver.  
Preparing to unpack .../3-gdbserver_9.2-0ubuntu1-20.04.1_amd64.deb ...  
Unpacking gdbserver (9.2-0ubuntu1-20.04.1) ...  
Selecting previously unselected package libcc1-0:amd64.
```

## Programme aktualisieren

Mit „apt upgrade“ können Programme aktualisiert werden. Dafür werden benötigte Pakete aufgelöst, die Download- und Installationsgröße bestimmt und erforderliche Pakete heruntergeladen und installiert. Meist ist kein Neustart nötig.

```
fische11@DESKTOP-SMHG39V: ~  
fische11@DESKTOP-SMHG39V:~$ sudo apt upgrade  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Calculating upgrade... Done  
The following package was automatically installed and is no longer required:  
  libfwdppplugin1  
Use 'sudo apt autoremove' to remove it.  
The following packages will be upgraded:  
  curl dbus dbus-user-session dbus-x11 libc-bin libcurl3-gnutls libcurl4 libdbus-1-3 libnss-systemd  
  libnss3 libpam-systemd libsensors-config libsensors5 libsqlite3-0 libssl1.1 libsystemd0 libtiff5  
  libudev1 locales networkd-dispatcher openssh-client openssh-server openssh-sftp-server openssl  
  rsyslog sbsigntool snapd systemd systemd-sysv systemd-timesyncd udev  
31 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
14 standard security updates  
Need to get 51.1 MB of archives.  
After this operation, 45.1 kB of additional disk space will be used.  
Do you want to continue? [Y/n]  
Get:1 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 locales all 2.31-0ubuntu9.9 [3869 kB]  
3% [1 locales 570 kB/3869 kB 15%]
```

Eine Aktualisierung kann auch mit „apt dist-update“ durchgeführt werden. Dieser Befehl funktioniert wie „apt update“. Hier werden neue Pakete installiert, wenn eine Abhängigkeit besteht. Außerdem kann so ein Kernel-Update und eine Aktualisierung auf die neue Distributionsversion durchgeführt werden.

```
fische11@DESKTOP-5MHG39V: ~
fische11@DESKTOP-5MHG39V:~$ sudo apt dist-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
fische11@DESKTOP-5MHG39V:~$
```

## Software deinstallieren

Mit „`apt remove <Programm>`“ kann Software deinstalliert werden. Hierbei werden nicht benötigte Pakete aufgelöst, die gelöschte Dateigröße bestimmt und entsprechende Pakete deinstalliert. Die Konfiguration hingegen bleibt bestehen.

```
fische11@DESKTOP-5MHG39V: ~
fische11@DESKTOP-5MHG39V:~$ sudo apt remove gdb
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gdbserver libbabeltrace1 libc6-dbg libcc1-0 libdw1 libfwupdplugin1
Use 'sudo apt autoremove' to remove them.
The following packages will be REMOVED:
  gdb
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.
After this operation, 9761 kB disk space will be freed.
Do you want to continue? [Y/n]
(Reading database ... 53772 files and directories currently installed.)
Removing gdb (9.2-0ubuntu1~20.04.1) ...
Processing triggers for man-db (2.9.1-1) ...
fische11@DESKTOP-5MHG39V:~$
```

Mit `apt purge <Programm>` kann ebenfalls Software deinstalliert werden. Der Befehl funktioniert wie `apt remove`, jedoch wird hier zusätzlich die Konfiguration gelöscht.

```
fische11@DESKTOP-5MHG39V: ~
fische11@DESKTOP-5MHG39V:~$ sudo apt purge gdb
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gdbserver libbabeltrace1 libc6-dbg libcc1-0 libdw1 libfwupdplugin1
Use 'sudo apt autoremove' to remove them.
The following packages will be REMOVED:
  gdb*
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.
After this operation, 9761 kB disk space will be freed.
Do you want to continue? [Y/n]
(Reading database ... 53772 files and directories currently installed.)
Removing gdb (9.2-0ubuntu1~20.04.1) ...
Processing triggers for man-db (2.9.1-1) ...
(Reading database ... 53694 files and directories currently installed.)
Purging configuration files for gdb (9.2-0ubuntu1~20.04.1) ...
fische11@DESKTOP-5MHG39V:~$
```

Weiterhin kann Software mit `apt autoremove` entfernt werden. Hierbei werden installierte Pakete automatisch gelöscht. Die Pakete ohne eine lokale Abhängigkeit, sprich Pakete ohne Verwendung, werden aufgelistet. Weiterhin wird die gelöschte Dateigröße bestimmt und die Pakete deinstalliert. Wie auch bei `apt remove` bleibt die Konfiguration bestehen.

```
fische11@DESKTOP-5MHG39V: ~
fische11@DESKTOP-5MHG39V:~$ sudo apt autoremove
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be REMOVED:
  gdbserver libbabeltrace1 libc6-dbg libcc1-0 libdw1 libfwupdplugin1
0 upgraded, 0 newly installed, 6 to remove and 0 not upgraded.
After this operation, 33.0 MB disk space will be freed.
Do you want to continue? [Y/n]
(Reading database ... 53692 files and directories currently installed.)
Removing gdbserver (9.2-0ubuntu1~20.04.1) ...
Removing libbabeltrace1:amd64 (1.5.8-1build1) ...
Removing libc6-dbg:amd64 (2.31-0ubuntu9.9) ...
Removing libcc1-0:amd64 (10.3.0-1ubuntu~20.04) ...
Removing libdw1:amd64 (0.176-1.1build1) ...
Removing libfwupdplugin1:amd64 (1.5.11-0ubuntu1~20.04.2) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...
fische11@DESKTOP-5MHG39V:~$
```

Auch mit apt autoclean kann Software deinstalliert werden. Hier werden die nicht gelisteten Pakete aufgelöst und Pakete deinstalliert. Meist wird dieser Befehl benutzt, um einen alten Linux-Kernel zu entfernen.

```
fische11@DESKTOP-5MHG39V: ~
fische11@DESKTOP-5MHG39V:~$ sudo apt autoclean
Reading package lists... Done
Building dependency tree
Reading state information... Done
fische11@DESKTOP-5MHG39V:~$
```

### Software auflisten

Um Software aufzulisten kann „apt list“ („apt list <Programm>“) verwendet werden. Es werden bekannte Pakete und der lokale Speicher aufgelistet. Um die Liste zu aktualisieren, verwendet man „apt update“.

```
fische11@DESKTOP-5MHG39V: ~
fische11@DESKTOP-5MHG39V:~$ sudo apt list gdb -a
Listing... Done
gdb/focal-updates 9.2-0ubuntu1~20.04.1 amd64
gdb/focal 9.1-0ubuntu1 amd64

fische11@DESKTOP-5MHG39V:~$ sudo apt list | nl -b a -n ln | tail -n 1 | cut -d \ -f 1
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

72090
fische11@DESKTOP-5MHG39V:~$ sudo apt list
Display all 72089 possibilities? (y or n)
fische11@DESKTOP-5MHG39V:~$ sudo apt list sha*
Listing... Done
shade-inventory/focal 1.31.0-0ubuntu1 all
shadowssocks-libev/focal 3.3.4+ds-2 amd64
shairport-sync/focal 3.3.5-1build1 amd64
shake/focal 1.0.2-1 all
shanty/focal 3-4 all
shapeit4-example/focal 4.1+dfsg-1build1 all
shapeit4/focal 4.1+dfsg-1build1 amd64
shapelib/focal 1.5.0-1build1 amd64
shapetools-tutorial/focal 1.3-5 all
shapetools/focal 1.4pl6-14 amd64
shared-mime-info/focal,now 1.15-1 amd64 [installed,automatic]
shariness/focal 1.0.0-1 all
sharutils-doc/focal 1:4.15.2-4build1 all
sharutils/focal 1:4.15.2-4build1 amd64
shatag/focal 0.5.0-3 all
fische11@DESKTOP-5MHG39V:~$
```



## Softwaredetails anzeigen

Mit „`apt show <Programm>`“ können Softwaredetails angezeigt werden. Hierbei werden Metadaten aufgelistet, wozu der Programmierer, die Webseite, die Version, die Größe, die Abhängigkeiten, die Beschreibung und der Paketspeicherort (Quelle) gehören. Mit der Option `-a` werden alle Versionen angezeigt.

```
fische11@DESKTOP-5MHG39V:~$ sudo apt show gdb -a
Package: gdb
Version: 9.2-0ubuntu1-20.04.1
Priority: optional
Section: devel
Origin: Ubuntu
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Héctor Orón Martínez <zumbi@debian.org>
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Installed-Size: 9761 kB
Provides: gdb-minimal (= 9.2-0ubuntu1-20.04.1)
Depends: libbabeltrace1 (>= 1.5.4), libc6 (>= 2.29), libexpat1 (>= 2.0.1), libgcc-s1 (>= 3.0), liblzma5 (>= 5.1.1alpha+20110809), libmpfr6 (>= 3.1.3), libncursesw6 (>= 6), libpython3.8 (>= 3.8.2), libreadline8 (>= 8), libstdc++6 (>= 7), libtinfo6 (>= 6), zlib1g (>= 1:1.2.0)
Recommends: libc-dbg, libcc1-0, gdbserver
Suggests: gdb-doc
Conflicts: gdb
Replaces: gdb, gdb-doc (<< 7.8-1~)
Homepage: http://www.gnu.org/s/gdb/
Task: ubuntu-desktop-minimal, ubuntu-desktop, kubuntu-desktop, xubuntu-desktop, ubuntustudio-desktop, ubuntu-kylin-desktop, ubuntu-mate-core, ubuntu-mate-desktop, ubuntu-budgie-desktop
Download-Size: 3222 kB
APT-Sources: http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages
Description: GNU Debugger
  GDB is a source-level debugger, capable of breaking programs at any specific line, displaying variable values, and determining where errors occurred. Currently, gdb supports C, C++, D, Objective-C, Fortran, Java, OpenCL C, Pascal, assembly, Modula-2, Go, and Ada. A must-have for any serious programmer.
Package: gdb
Version: 9.1-0ubuntu1
Priority: optional
Section: devel
Origin: Ubuntu
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Héctor Orón Martínez <zumbi@debian.org>
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Installed-Size: 9759 kB
Provides: gdb-minimal (= 9.1-0ubuntu1)
Depends: libbabeltrace1 (>= 1.5.4), libc6 (>= 2.29), libexpat1 (>= 2.0.1), libgcc-s1 (>= 3.0), liblzma5 (>= 5.1.1alpha+20110809), libmpfr6 (>= 3.1.3), libncursesw6 (>= 6), libpython3.8 (>= 3.8.0-0a1), libreadline8 (>= 8), libstdc++6 (>= 7), libtinfo6 (>= 6), zlib1g (>= 1:1.2.0)
Recommends: libc-dbg, libcc1-0, gdbserver
Suggests: gdb-doc
Conflicts: gdb
Replaces: gdb, gdb-doc (<< 7.8-1~)
Homepage: http://www.gnu.org/s/gdb/
Task: ubuntu-desktop-minimal, ubuntu-desktop, kubuntu-desktop, xubuntu-desktop, ubuntustudio-desktop, ub
```

## Weitere Repositories hinzufügen

Die Repositories werden in „`/etc/apt/sources.list`“ bzw. in `/etc/apt/sources.list.d/*`“ aufgelistet. Man kann sie manuell einschreiben mit dem Befehl „`add-apt-repository <Repository>`“. Dabei bekommt man Informationen zum Softwarehersteller. Wichtig ist, dass man für die Integritätsprüfung den öffentlichen Schlüssel (public key) importieren muss. Dies erfolgt mit „`apt-key`“.

```

fische11@DESKTOP-5MKG39V: ~
fische11@DESKTOP-5MKG39V:~$ sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 90A316203348D75D9DCB49F368818C72E52529D4
Executing: /tmp/apt-key-gpghome.Yf1c9JFM36/gpg.1.sh --keyserver hkp://keyserver.ubuntu.com:80 --recv 90A316203348D75D9DCB49F368818C72E52529D4
gpg: key 68818C72E52529D4: public key "MongoDB 4.0 Release Signing Key <packaging@mongodb.com>" imported
gpg: Total number processed: 1
gpg:      imported: 1
fische11@DESKTOP-5MKG39V:~$ sudo add-apt-repository 'deb [arch=amd64] https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.0 multiverse'
Ign:1 https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.0 InRelease
Get:2 https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.0 Release [2989 B]
Get:3 https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.0 Release.gpg [801 B]
Hit:4 http://archive.ubuntu.com/ubuntu focal InRelease
Get:5 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:7 https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.0/multiverse amd64 Packages [18.4 kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [1451 kB]
Get:9 http://archive.ubuntu.com/ubuntu focal-backports InRelease [188 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1790 kB]
Get:11 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [921 kB]
Fetched 4519 kB in 2s (2736 kB/s)
Reading package lists... Done
fische11@DESKTOP-5MKG39V:~$ cat /etc/apt/sources.list | head -n 20
# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.
deb http://archive.ubuntu.com/ubuntu/ focal main restricted
deb-src http://archive.ubuntu.com/ubuntu/ focal main restricted

## Major bug fix updates produced after the final release of the
## distribution.
deb http://archive.ubuntu.com/ubuntu/ focal-updates main restricted
deb-src http://archive.ubuntu.com/ubuntu/ focal-updates main restricted

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team. Also, please note that software in universe WILL NOT receive any
## review or updates from the Ubuntu security team.
deb http://archive.ubuntu.com/ubuntu/ focal universe
deb-src http://archive.ubuntu.com/ubuntu/ focal universe
deb http://archive.ubuntu.com/ubuntu/ focal-updates universe
deb-src http://archive.ubuntu.com/ubuntu/ focal-updates universe

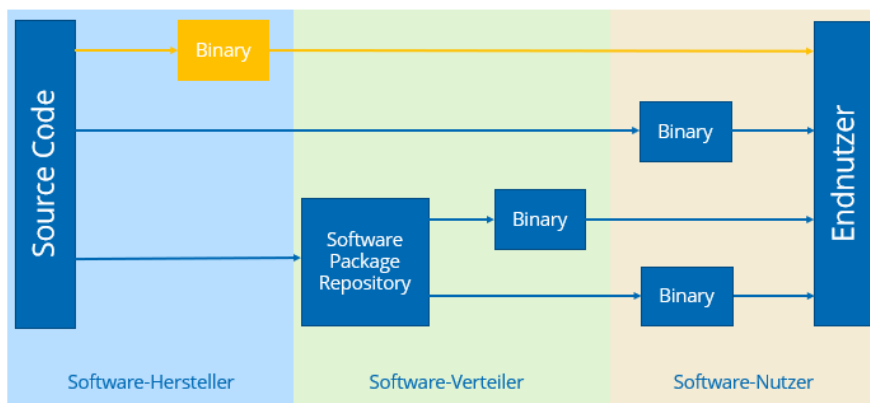
## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
fische11@DESKTOP-5MKG39V:~$ cat /etc/apt/sources.list | tail -n 2
deb [arch=amd64] https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.0 multiverse
deb-src [arch=amd64] https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.0 multiverse
fische11@DESKTOP-5MKG39V:~$

```

## Softwareverwaltung mit Skripten

Für die Softwareverwaltung mit Skripten kann man einen Arbeitsplatz vorbereiten. Dies geschieht mit „apt update -y“ bzw. „apt install -y <Programm1> <Programm2>...“. Somit lässt sich die Installation von Software automatisieren. Dieser Arbeitsplatz ist mit hauseigenen Repositories erweiterbar. Weiterhin kann man mit Skripten eine automatisierte Aktualisierung durchführen. Einige Skripte sind im Cron-Dienst hinterlegt. Dazu zählen „apt update -y“, „apt upgrade -y“, „apt dist-upgrade -y“, „apt autoremove -y“ und „apt autoclean -y“. Damit lässt sich eine Automatisierung der Softwareupdates durchführen. Weiterhin gibt es „unattended-upgrades“, womit automatisierte Updates über „apt“ laufen.

### 1.6.3 Binaries



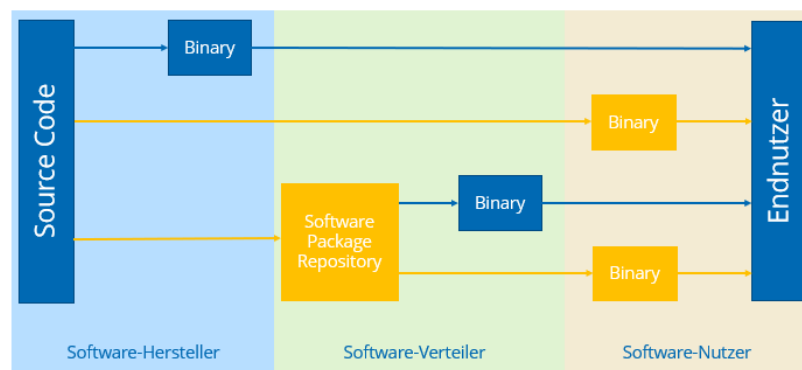
### 1.6.3.1 Binary

Die Datei heißt Binary, da sie nicht menschenlesbar ist. Bei der Binary handelt es sich um ein ausführbares Programm, aber dessen Funktionsweise ist nicht überprüfbar. Somit benötigt man ein vollständiges Vertrauen in den Hersteller. Der Updateprozess läuft einzeln für jedes Programm. Bei dem Format handelt es sich hier um ein Applmage, welches früher auch als klik bzw. Portable Linux Apps bezeichnet wurde.

### 1.6.3.2 Applmage

Das Programm wird im Applmage Format heruntergeladen. Gegebenenfalls muss das Programm erst mit „chmod a+x <Programm>“ ausführbar gemacht werden. Der Start erfolgt über die Konsole mit „./Programm“. Soll der Start über den Dateimanager erfolgen, so wählt man das Programm mit einem Doppelklick aus. Alle Abhängigkeiten sind im Applmage Format bereits integriert.

### 1.6.4 Source Code



Der Source Code ist der Quelltext, wobei es sich um menschenlesbare Programmanweisungen handelt. Eine Übersetzung in maschinenlesbare Anweisungen ist hier notwendig, welche mit einmaligem Kompilieren und Interpretieren während der Ausführung umgesetzt wird. Den Source Code kann man auf verschiedenste Weise downloaden. Beispielsweise über Webseiten wie github, gitlab oder die Herstellerwebseite. Über git kann man den Code auch mit dem Befehl „git clone <Repository>“ herunterladen. Mittels apt erfolgt der Download über „apt source <Programm>“. Eine Installationsanleitung befindet sich entweder in der README.MD oder der INSTALL.MD.

### 1.6.4.1 Aktualisierung mit Source Code

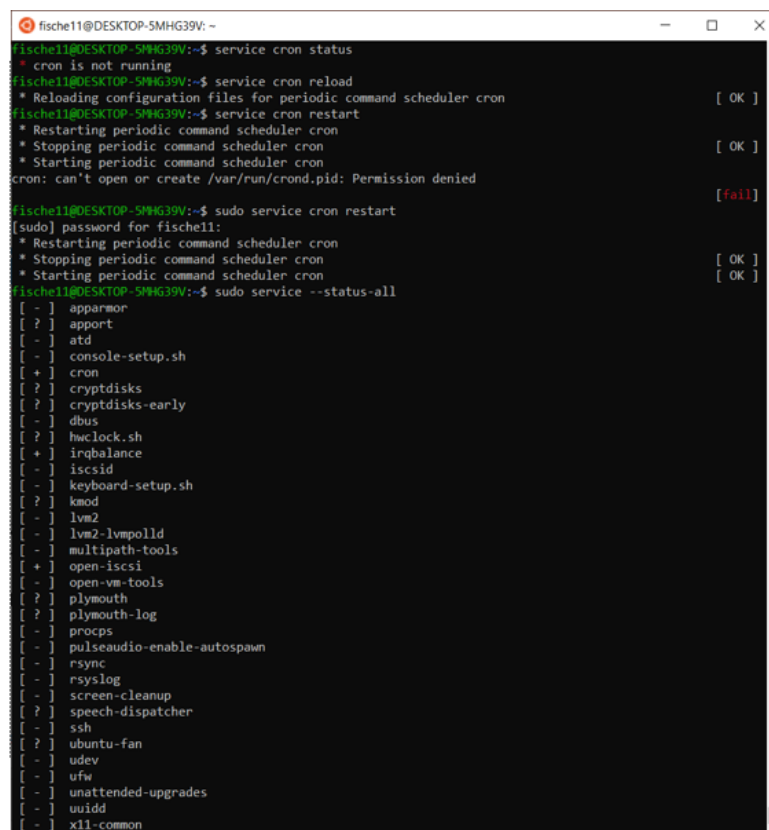
Für die Aktualisierung wird die alte Version manuell entfernt und der Source Code erneut heruntergeladen. Wenn man den Code über ein git-repository geclont hat, verwendet man den Befehl „git pull“.

Generell ist ein erneutes Kompilieren notwendig, um den Code in Maschinencode zu übersetzen. So eine Aktualisierung des Source Codes ist aufwendig, daher sollte wenn möglich ein Paketmanager verwendet werden.

## 1.6.5 Dienstverwaltung

### 1.6.5.1 Hintergrunddienste (Deamons)

Hintergrunddienste werden mit dem Befehl „service“ verwaltet. Mit „status“ wird der aktuelle Status ausgelesen, mit „start“ ein Service gestartet, mit „stop“ ein Service angehalten, mit „reload“ ein Service zum Auslesen der Konfiguration aufgefordert und mit „restart“ ein Service neu gestartet.



```
fische11@DESKTOP-5PHG39V: ~$ service cron status
* cron is not running
fische11@DESKTOP-5PHG39V: ~$ service cron reload
* Reloading configuration files for periodic command scheduler cron [ OK ]
fische11@DESKTOP-5PHG39V: ~$ service cron restart
* Restarting periodic command scheduler cron
* Stopping periodic command scheduler cron [ OK ]
* Starting periodic command scheduler cron
cron: can't open or create /var/run/crond.pid: Permission denied [fail]
fische11@DESKTOP-5PHG39V: ~$ sudo service cron restart
[sudo] password for fische11:
* Restarting periodic command scheduler cron
* Stopping periodic command scheduler cron [ OK ]
* Starting periodic command scheduler cron [ OK ]
fische11@DESKTOP-5PHG39V: ~$ sudo service --status-all
[ - ] apparmor
[ ? ] appport
[ - ] atd
[ - ] console-setup.sh
[ + ] cron
[ ? ] cryptdisks
[ ? ] cryptdisks-early
[ - ] dbus
[ ? ] hwclock.sh
[ + ] irqbalance
[ - ] iscsid
[ - ] keyboard-setup.sh
[ ? ] kmod
[ - ] lvm2
[ - ] lvm2-lvmpolld
[ - ] multipath-tools
[ + ] open-iscsi
[ - ] open-vm-tools
[ ? ] plymouth
[ ? ] plymouth-log
[ - ] procs
[ - ] pulseaudio-enable-autospawn
[ - ] rsync
[ - ] rsyslog
[ - ] screen-cleanup
[ ? ] speech-dispatcher
[ - ] ssh
[ ? ] ubuntu-fan
[ - ] udev
[ - ] ufw
[ - ] unattended-upgrades
[ - ] uuidd
[ - ] x11-common
```

## 1.6.6 Zusammenfassung

Sie kennen nun alle verschiedene Möglichkeiten Software unter Linux zu installieren. Insbesondere wurde ein besonderes Augenmerk auf die Verwendung von Paketmanagern verwendet.

Hier wurde am Beispiel des Paketmanagers apt die Installation, Aktualisierung und Deinstallation von Programmen erläutert.

Als Alternativen zum Paketmanager wurden Binaries über das Appliance-Format und selbstkomprimierte Programme aus Source Code.

Am Ende lernten Sie, wie laufende Hintergrunddienste unter Linux mit dem service-Befehl verwaltet werden.

## 1.7 Linux Selbsthilfe

### 1.7.1 Manual Pages

Linux User sind tendenziell hilfsbereit, aber genervt von lesefaulen Nutzern. Aufgrund dessen hört man oft den Spruch „Read The Fucking Manual“ (Lese das verdammte Handbuch), kurz RFTM. Fast jedes Programm hat heute eine Bedienungsanleitung (manual). Dieses ruft man über „man <Programm>“ ab.

A screenshot of a Linux terminal window. The window title is 'fische11@DESKTOP-5MHG39V: -'. The terminal prompt is 'fische11@DESKTOP-5MHG39V:~\$' and the command 'man cut' has been entered. The terminal output is mostly black, indicating that the manual page content is not visible in this view.

#### 1.7.1.1 Aufbau Manpages

Die Manpages sind alle auf dieselbe Art aufgebaut. Zuerst wird der Name des Programms mit einer Kurzbeschreibung in wenigen Wörtern genannt. Daraufhin folgt die Synopsis, sprich die Befehlssyntax mit erforderlichen und optionalen Elementen. In der Description wird das Programm und deren Einsatzzweck beschrieben. Im nächsten Abschnitt folgen die Optionen, womit Konfigurationsmöglichkeiten des Programms gemeint sind. Daraufhin kommt der Author, wobei es sich um den Programmierer bzw. den Verantwortlichen handelt. Dann folgen Reporting Bugs (Programmfehlermeldestelle), Copyright/License für die Verteilung bzw. die Abänderung, See also mit weiteren Infos und Expamples, wo Beispiele genannt werden. Diese sind leider nicht immer vorhanden oder auch nicht immer ausführlich.

```
fische11@DESKTOP-5MHG39V: ~
DF(1) User Commands DF(1)
NAME
df - report file system disk space usage

SYNOPSIS
df [OPTION]... [FILE]...

DESCRIPTION
This manual page documents the GNU version of df. df displays the amount of disk space available on the file system containing each file name argument. If no file name is given, the space available on all currently mounted file systems is shown. Disk space is shown in 1K blocks by default, unless the environment variable POSIXLY_CORRECT is set, in which case 512-byte blocks are used.

If an argument is the absolute file name of a disk device node containing a mounted file system, df shows the space available on that file system rather than on the file system containing the device node. This version of df cannot show the space available on unmounted file systems, because on most kinds of systems doing so requires very nonportable intimate knowledge of file system structures.

OPTIONS
Show information about the file system on which each FILE resides, or all file systems by default.

Mandatory arguments to long options are mandatory for short options too.

-a, --all
    include pseudo, duplicate, inaccessible file systems

-B, --block-size=SIZE
    scale sizes by SIZE before printing them; e.g., '-BM' prints sizes in units of 1,048,576 bytes; see SIZE format below

-h, --human-readable
    print sizes in powers of 1024 (e.g., 1023M)

-H, --si
    print sizes in powers of 1000 (e.g., 1.1G)

-i, --inodes
    list inode information instead of block usage

-k like --block-size=1K

-l, --local
    limit listing to local file systems

Manual page df(1) line 1/108 45% (press h for help or q to quit)
```

### 1.7.1.2 Kategorien

Es gibt verschiedene Kategorien, in die die Manual Pages unterteilt sind. Dazu zählen folgende zehn Kategorien:

- 1 Benutzerkommandos, z.B. ls
- 2 Systemfunktionen, z.B. fork()
- 3 Subroutinen in Bibliotheken
- 4 Geräte(treiber) z.B. von XFree86
- 5 Dateiformate, z.B. hosts, passwd
- 6 Spiele, z.B. gnuchess
- 7 Verschiedenes, z.B. man
- 8 Systemverwaltung (Befehle für root), z.B. ip, sudo
- 9 Kernel
- n neu

```
fische11@DESKTOP-5MHG39V: ~
fische11@DESKTOP-5MHG39V:~$ whatis git
Git (3pm)          - Perl interface to the Git version control system
git (1)           - the stupid content tracker
fische11@DESKTOP-5MHG39V:~$ man git | head -n 10
GIT(1)            Git Manual                                GIT(1)

NAME
  git - the stupid content tracker

SYNOPSIS
  git [--version] [--help] [-C <path>] [-c <name>=<value>]
    [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
    [-p|--paginate|-P|--no-pager] [--no-replace-objects] [--bare]
    [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
fische11@DESKTOP-5MHG39V:~$ man git.1 | head -n 10
GIT(1)            Git Manual                                GIT(1)

NAME
  git - the stupid content tracker

SYNOPSIS
  git [--version] [--help] [-C <path>] [-c <name>=<value>]
    [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
    [-p|--paginate|-P|--no-pager] [--no-replace-objects] [--bare]
    [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
fische11@DESKTOP-5MHG39V:~$ man git.3 | head -n 10
Git(3)           User Contributed Perl Documentation       Git(3)

NAME
  Git - Perl interface to the Git version control system

SYNOPSIS
  use Git;

  my $version = Git::command_online('version');
fische11@DESKTOP-5MHG39V:~$
```

### 1.7.1.3 Man Pages im Web

## Linux manual pages: alphabetic list of all pages

Jump to letter: . 3 a b c d e f g h i j k l m n o p q r s t u v w x y z

- top
  - [.ldaprc\(5\)](#) - LDAP configuration file/environment variables
- top
  - [30-systemd-environment-d-generator\(8\)](#) - Load variables specified by environment.d
- top
  - [a64l\(3\)](#) - convert between long and base-64
  - [a64l\(3p\)](#) - bit integer and a radix-64 ASCII string
  - [abicompat\(1\)](#) - check ABI compatibility
  - [abidiff\(1\)](#) - compare ABIs of ELF files
  - [abidw\(1\)](#) - serialize the ABI of an ELF file
  - [abilint\(1\)](#) - validate an abigail ABI representation
  - [abipkgdiff\(1\)](#) - compare ABIs of ELF files in software packages
  - [abort\(3\)](#) - cause abnormal process termination
  - [abort\(3p\)](#) - generate an abnormal process abort
  - [abs\(3\)](#) - compute the absolute value of an integer
  - [abs\(3p\)](#) - return an integer absolute value
  - [ac\(1\)](#) - print statistics about users' connect time
  - [accept\(2\)](#) - accept a connection on a socket
  - [accept\(3p\)](#) - accept a new connection on a socket
  - [accept4\(2\)](#) - accept a connection on a socket
  - [access\(2\)](#) - check user's permissions for a file
  - [access\(3p\)](#) - determine accessibility of a file descriptor
  - [access.conf\(5\)](#) - the login access control table file
  - [accessdb\(8\)](#) - dumps the content of a man-db database in a human readable format
  - [acct\(2\)](#) - switch process accounting on or off
  - [acct\(5\)](#) - process accounting file
  - [accton\(8\)](#) - turns process accounting on or off
  - [acl\(5\)](#) - Access Control Lists



## gdb(1) — Linux manual page

[NAME](#) | [SYNOPSIS](#) | [DESCRIPTION](#) | [OPTIONS](#) | [SEE ALSO](#) | [COPYRIGHT](#) | [COLOPHON](#)

GDB(1) GNU Development Tools GDB(1)

### NAME [top](#)

gdb - The GNU Debugger

### SYNOPSIS [top](#)

```
gdb [-help] [-nh] [-nx] [-q] [-batch] [-cd=dir] [-f] [-b bps]
    [-tty=dev] [-s symfile] [-e prog] [-se prog] [-c core]
    [-p procID]
    [-x cmds] [-d dir] [prog|prog procID|prog core]
```

### DESCRIPTION [top](#)

The purpose of a debugger such as GDB is to allow you to see what is going on "inside" another program while it executes -- or what another program was doing at the moment it crashed.

GDB can do four main kinds of things (plus other things in support of these) to help you catch bugs in the act:

- Start your program, specifying anything that might affect its behavior.

### 1.7.1.4 info

„info <Befehl>“ liefert Informationen zu GNU-utils. Die Informationen sind wie ein großes Buch aufgebaut. info wird nur für den „Buchstart“ genutzt. Hier finden sich viele Informationen zu Linux. Unterstrichene Texte sind Links, welchen man über die Enter-Taste folgen kann. Zurück kommt man mit der Backspace-Taste.

```
fische11@DESKTOP-5MHG39V: ~
Next: Symbolic Modes, Up: File permissions

27.1 Structure of File Mode Bits
=====

The file mode bits have two parts: the "file permission bits", which
control ordinary access to the file, and "special mode bits", which
affect only some files.

There are three kinds of permissions that a user can have for a file:

1. permission to read the file. For directories, this means
   permission to list the contents of the directory.
2. permission to write to (change) the file. For directories, this
   means permission to create and remove files in the directory.
3. permission to execute the file (run it as a program). For
   directories, this means permission to access files in the
   directory.

There are three categories of users who may have different
permissions to perform any of the above operations on a file:

1. the file's owner;
2. other users who are in the file's group;
3. everyone else.

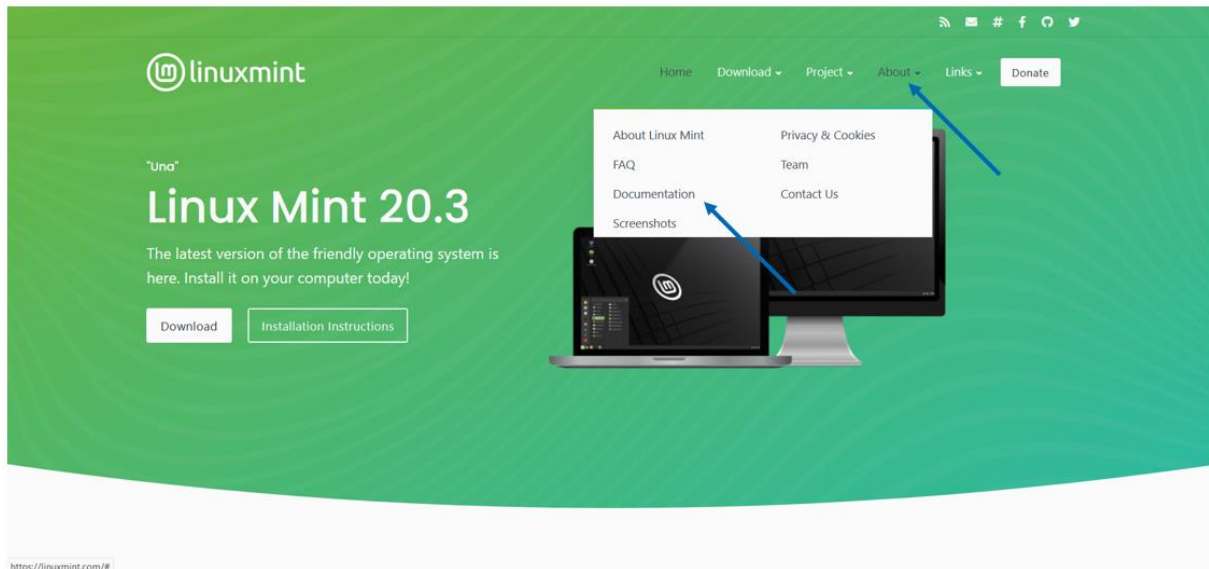
Files are given an owner and group when they are created. Usually
the owner is the current user and the group is the group of the
directory the file is in, but this varies with the operating system, the
file system the file is created on, and the way the file is created.
You can change the owner and group of a file by using the 'chown' and
'chgrp' commands.

In addition to the three sets of three permissions listed above, the
file mode bits have three special components, which affect only
executable files (programs) and, on most systems, directories:

1. Set the process's effective user ID to that of the file upon
   execution (called the "set-user-ID bit", or sometimes the "setuid
   bit"). For directories on a few systems, give files created in the
   directory the same owner as the directory, no matter who creates
   them, and set the set-user-ID bit of newly-created subdirectories.
2. Set the process's effective group ID to that of the file upon
   execution (called the "set-group-ID bit", or sometimes the "setgid
   bit"). For directories on most systems, give files created in the
   directory the same group as the directory, no matter what group the
   user who creates them is in, and set the set-group-ID bit of
   -----
Info: (coreutils)Mode Structure, 82 lines --top
```

## 1.7.2 Herstellerwebseite

### 1.7.2.1 Linux Mint



## Installations-Guide

Für die verschiedenen Kategorien werden Anleitungen bereitgestellt. Der Installations-Guide ist sehr detailliert und sogar mehrsprachig vorhanden. Zu beachten ist, dass manche Sätze nicht übersetzt werden. Außerdem erhält man ausführliche Erklärungen

**First Read**

## Installation Guide

The Linux Mint Installation Guide helps you download the right ISO image, create your bootable media and install Linux Mint on your computer.

It is available in **HTML, PDF and ePub** in the following languages:

English	French
Amharic	German
Arabic	Greek
Basque	Interlingua
Bulgarian	Italian
Catalan	Korean
Chinese	Lithuanian
Croatian	Portuguese (Brazil)
Czech	Russian
Danish	Spanish
Dutch	Swedish
Esperanto	Turkish



Docs » Den startfähigen Datenträger erstellen

[Edit on GitLab](#)

## Den startfähigen Datenträger erstellen

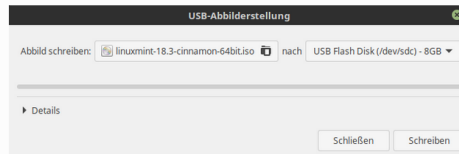
Der einfachste Weg, Linux Mint zu installieren, ist mit einem USB-Stick.

Wenn das Starten vom USB-Stick nicht funktioniert, können Sie eine leere DVD benutzen.

### Wie man einen startfähigen USB-Stick erstellt

#### In Linux Mint

Rechtsklick auf die ISO-Datei und »Startfähigen USB-Stick erstellen« auswählen oder Menü »Zubehör » USB-Abbilderstellung öffnen.

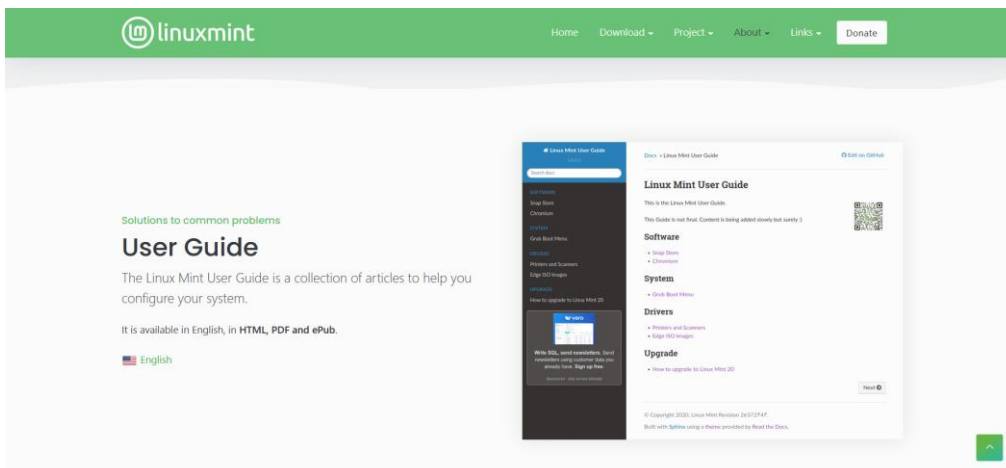


Wählen Sie Ihren USB-Stick aus und klicken Sie auf [Schreiben](#).

#### In Windows, Mac OS oder anderen Linux-Distributionen

## User Guide

Der User Guide enthält weniger Informationen als der Installations-Guide. Dafür werden einzelne spezielle Themen angesprochen, von welchen die wichtigsten Abschnitte thematisiert werden. Dennoch ist der User Guide sehr unvollständig.



Linux Mint User Guide  
latest

Search docs

**SOFTWARE**  
Snap Store  
Chromium

**SYSTEM**

Grub Boot Menu

- How to make the Grub menu always visible
- How to theme the Grub menu

**DRIVERS**  
Printers and Scanners  
Edge ISO Images

**UPGRADE**  
Upgrades  
How to upgrade to Linux Mint 20

Read the Docs v: latest

## Grub Boot Menu

Grub is the boot menu.

If you have more than one operating system installed, it allows you to select which one to boot.

Grub is also useful for troubleshooting. You can use it to modify the boot arguments or to boot from an older kernel.

### How to make the Grub menu always visible

If you only run Linux Mint and there are no other operating systems on the computer, the menu is hidden by default.

To make it visible, as root, add these lines to `/etc/default/grub.d/90_custom.cfg`:

```
GRUB_TIMEOUT="--5"
GRUB_TIMEOUT_STYLE="menu"
```

Then type the following commands in a terminal:

```
sudo update-grub
```

### How to theme the Grub menu

For compatibility reasons, some releases sometimes ship without a Grub theme:

## Fazit

Herstellerwebseiten wie Linux Mint sind gut geeignet für die Installation, der Rest ist jedoch ausbaufähig. Weiterhin findet man hier aber auch Links zu Foren und Social-Media als weitere Anlaufstelle.

### 1.7.2.2 gnu.org

gnu.org bietet eine Übersicht über GNU-Tools sowie ausführliche Handbücher für viele Konsolenbefehle.

Den Free Software Supporter abonnieren: E-Mail-Adresse

**GNU Betriebssystem**  
Gefördert von der Free Software Foundation

ÜBER GNU PHILOSOPHIE LIZENZEN BILDUNG SOFTWARE DISTROS **DOKUMENTATION** SCHADSOFTWARE

Diese Übersetzung berücksichtigt möglicherweise nicht mehr die seit 2022-01-19 gemachten Änderungen der englischsprachigen Originalfassung (die Unterschiede). Wenden Sie sich bitte unter [eww@de-translators](#) an das Übersetzungsteam, wenn Sie mithelfen möchten diese Übersetzung zu aktualisieren.

## Dokumentation des GNU-Projekts

GNU- und andere freie Dokumentation sind mithilfe folgender Methoden erhältlich:

- **GNU-Handbücher** sind meistens in verschiedenen Formaten online:
  - **Software-Verzeichnis**: enthält Verweise zu Dokumentationen;
  - **et GNU**: Blogs und Neuigkeiten von GNU-Paketen;
  - **Onlineshop**: Bücher, T-Shirts, Poster und vieles mehr;
  - **Bücher von anderen Verlegern**: Bücher, die freie Dokumentationslizenzen verwenden.

### Dokumentationsprinzipien

Die Dokumentation wurde ursprünglich unter einer Lizenz mit **Copyright** in Kurzform oder unter der **GNU General Public License (GPL)** freigegeben. Im Jahr 2001 wurde die Dokumentation unter der **GNU Free Documentation License (FDL)** freigegeben, um Bedürfnisse abzudecken, die nicht durch ursprünglich für Software entworfene Lizenzen erfüllt werden können. Für weitere Informationen über freie Dokumentationen siehe den Aufsatz **Freie Software und freie Handbücher** von Richard Stallman (eine Variante dieses Aufsatzes ist auch für die Aufnahme in Handbüchern als **Texinfo-Quelldatei** abrufbar).

**GNU Gzip: General file (de)compression**

This manual is for GNU Gzip (version 1.12, 2 April 2022), and documents commands for compressing and decompressing data.

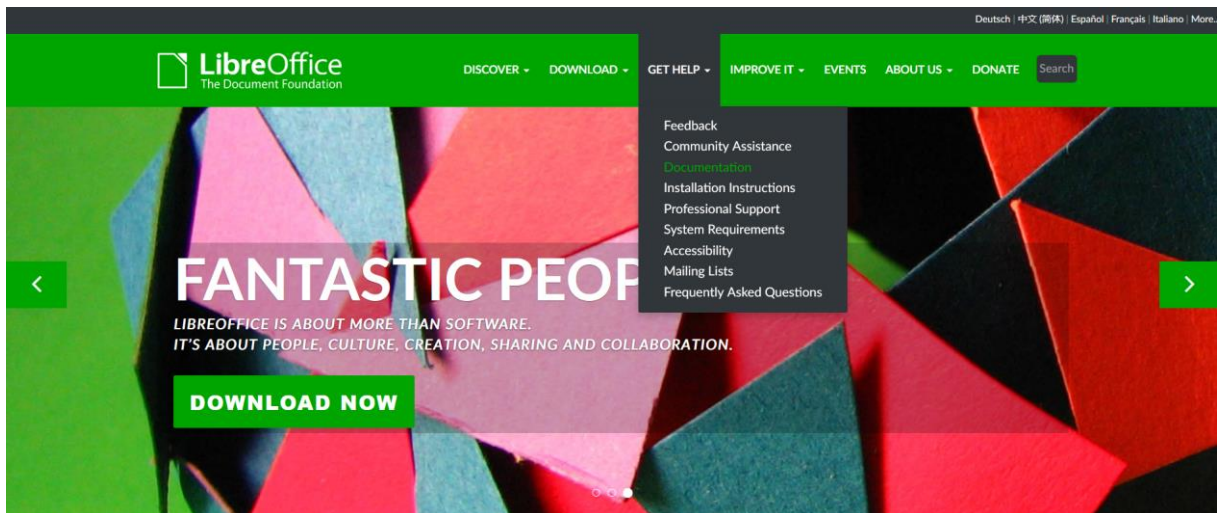
Copyright © 1998–1999, 2001–2002, 2006–2007, 2009–2022 Free Software Foundation, Inc.  
Copyright © 1992, 1993 Jean-loup Gailly

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

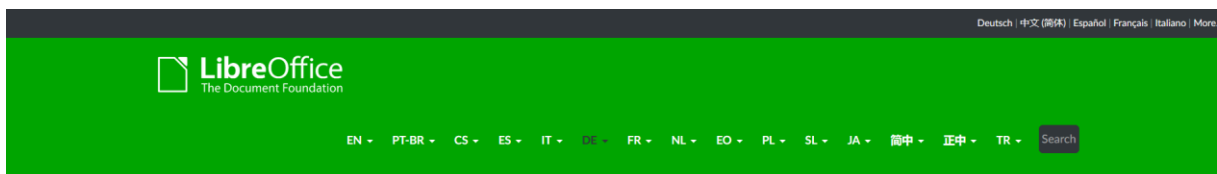
**Table of Contents**

- 1 Overview
- 2 Sample output

### 1.7.2.3 Libre Office



Die Handbücher von LibreOffice bieten eine ausführliche Dokumentation und sind mehrsprachig. Auf Englisch findet man die neueste und vollständigste Dokumentation.



#### Handbuch „Erste Schritte“:

Titel	Aktuell	Vorversionen
Kapitel 01 – Einführung in LibreOffice	6.2 ODT   PDF	6.0 ODT   PDF 5.2 ODT   PDF 3.5 ODT   PDF 3.3 ODT   PDF
Kapitel 02 – LibreOffice Einstellungen	6.2 ODT   PDF	6.0 ODT   PDF 5.2 ODT   PDF 3.3 ODT   PDF
Kapitel 03 – Einführung in Formate und Vorlagen	6.2 ODT   PDF	6.0 ODT   PDF 5.2 ODT   PDF 3.3 ODT   PDF
Kapitel 04 – Einführung in Writer	6.2 ODT   PDF	6.0 ODT   PDF 5.2 ODT   PDF 3.3 ODT   PDF
		6.0 ODT   PDF

### 1.7.2.4 Fazit

Die Software-Dokumentation ist sehr abhängig vom Hersteller, sodass auch die Qualität variiert. Libre Office ist hier ein positives Beispiel, da die Dokumentation sehr umfangreich, mehrsprachig, kostenlos und für alle Versionen verfügbar ist.

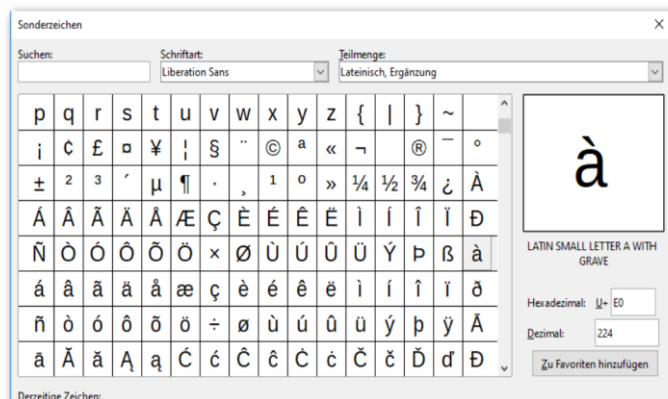
## Sonderzeichen einfügen

Die Tastatur bietet nur einen kleinen, wenn auch für die meisten Aufgaben ausreichenden Satz an Zeichen. Die weiteren im Schriftsatz verfügbaren Zeichen, die nicht direkt über die Tastatur einzugeben sind, werden als Sonderzeichen bezeichnet.

Für einige häufig benötigte Fälle gibt es Tastaturkombinationen oder Automatismen, z.B.

- Verwendung der Akzenttasten  $\grave{`}$ ,  $\acute{´}$  mit anschließendem Vokal, um akzentuierte Vokale im Französischen, Italienischen, ... einzugeben
- einfaches Tippen von "1/2", um  $\frac{1}{2}$  zu erzeugen (entsprechend funktioniert „1/4“ und „3/4“)
- die Tastenfolge „(c)“, um © zu erzeugen

In vielen Fällen gibt es eine solche Abkürzung nicht, oder Sie kennen sie nicht. Dann können Sie das Sonderzeichen, das Sie benötigen, über den Dialog *Sonderzeichen* (Abbildung 7) setzen.



## 1.7.3 Foren und Wikis

### 1.7.3.1 wikipedia.org

Unter wikipedia.org finden sich Wikipedia-Artikel für Programme und Distributionen. Hier bekommt man eher allgemeine Informationen anstatt einer Direkthilfe.

**dd (Unix)**

**dd** ist ein Unix-Kommando, das zum blockorientierten Kopieren oder Konvertieren beliebiger Dateien dient. Die Größe jedes „Datenblocks“ liegt üblicherweise zwischen einem einzigen Byte und Vielfachen der Blockgröße eines Dateisystems.

**Inhaltsverzeichnis** [Verbergen]

- 1 Geschichte
- 2 Arbeitsweise
  - 2.1 Konvertierung
  - 2.2 Statusanzeige
- 3 Beispiele
  - 3.1 Sichern und Wiederherstellen einer Partition
  - 3.2 Datenträger unlesbar machen
  - 3.3 Sparse-Datei erzeugen
- 4 Siehe auch
- 5 Weblinks
- 6 Anmerkungen

**Geschichte** [ Bearbeiten | Quelltext bearbeiten ]

Das Dienstprogramm **dd** war schon in den 1970er Jahren im Umlauf und hat sich seitdem unter Unix-Betriebssystemen als unverzichtbar erwiesen.<sup>[1]</sup> Als wichtiges Standard-Werkzeug ist das Programm Teil der core utilities des GNU-Projekts geworden. Zur Datenrettung stehen verschiedene Weiterentwicklungen unter der GPL wie z. B. *gddrescue*<sup>[2]</sup> zur Verfügung, die versuchen, trotz auftretender Lesefehler möglichst viele Daten wiederherzustellen.<sup>[3]</sup> Zur Datensicherung, -Wiederherstellung und -Rettung sowie Aufgaben der Forensik entstanden weitere Programme wie *actiadm*<sup>[4]</sup> und *ac3dm*<sup>[5]</sup> die auf dem Konzept von **dd** aufbauen und es erweitern.<sup>[6]</sup> Für Windows stehen verschiedene Open-Source-Portierungen von **dd** zur Verfügung, bspw. von Cygwin und chrysooome.net.<sup>[7]</sup>

Die Herkunft des Programmnamens **dd** ist nicht gänzlich geklärt und es finden sich zahlreiche Möglichkeiten in diversen Quellen:


- In der Job Control Language (JCL) der IBM-Großrechner S/360 gibt es die Anweisung **dd** für „Dataset Definition“.<sup>[8]</sup>
- Ursprünglich soll **cc** als Abkürzung für „copy and convert“ geplant gewesen sein, diese war aber schon für den C-Compiler vergeben.<sup>[9]</sup>
- Darüber hinaus kann **dd** als „duplicate data“, „disk dump“, „data dump“, „duplicate device“, „direct disc“ oder einer anderen Kombinationen von zwei dieser Wörter verstanden werden.<sup>[8][10]</sup>

Ironische Zuschreibung des Akronymys

- Wegen der häufigen Verwendung zum hardwarenahen Zugriff auf die Geräteschnittstellen von Festplatten oder deren Partitionen wird das Akronym auch als „destroy disk“ oder „delete data“ ausbuchstabiert, weil Tippfehler bei der Angabe der Ein- oder Ausgabegeräte zu vollständigem Datenverlust führen können.<sup>[1]</sup>



# Ubuntu

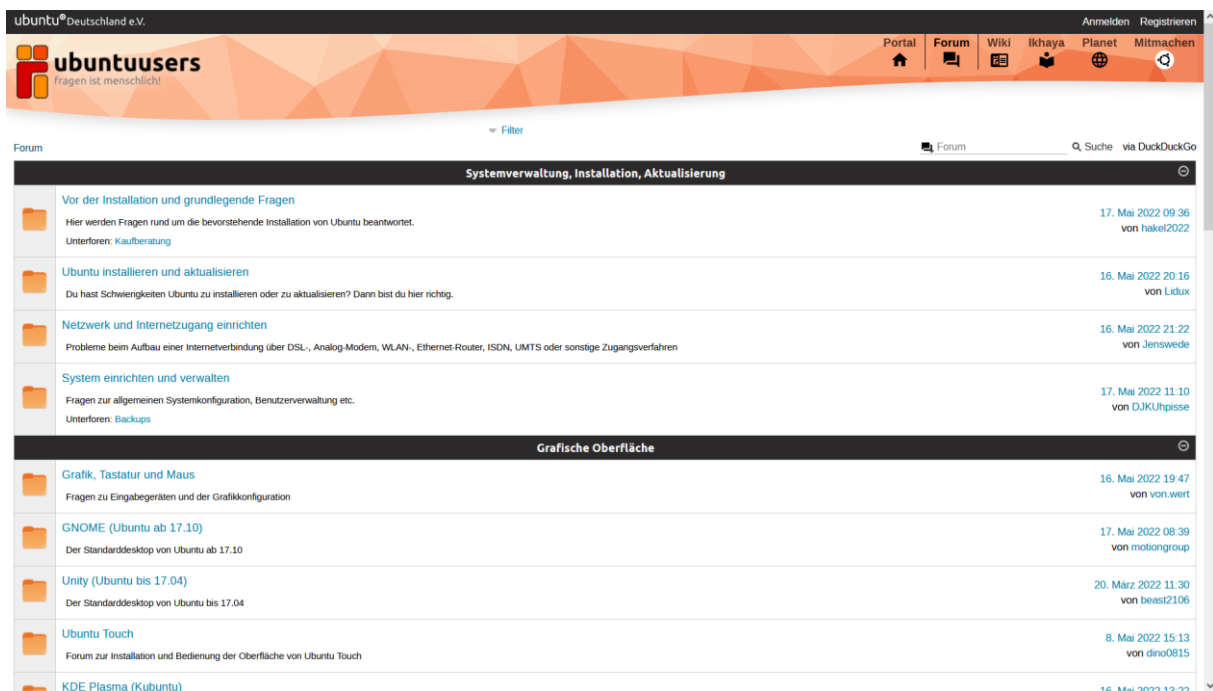
 Dieser Artikel behandelt die Linux-Distribution. Zu weiteren Bedeutungen siehe [Ubuntu \(Begriffsklärung\)](#).

**Ubuntu**, auch **Ubuntu Linux**, ist eine GNU/Linux-Distribution, die auf [Debian](#) basiert. Der Name Ubuntu bedeutet auf [Zulu](#) etwa „Menschlichkeit“ und bezeichnet eine [afrikanische Philosophie](#). Die Entwickler verfolgen mit dem System das Ziel, ein einfach zu installierendes und leicht zu bedienendes [Betriebssystem](#) mit aufeinander abgestimmter Software zu schaffen. Das Projekt wird vom Software-Hersteller [Canonical](#) gesponsert, der vom [südafrikanischen](#) Unternehmer [Mark Shuttleworth](#) gegründet wurde.<sup>[2]</sup>

Ubuntu konnte seit dem Erscheinen der ersten Version im Oktober 2004 seine Bekanntheit stetig steigern und ist inzwischen eine der meistgenutzten GNU/Linux-Distributionen. Die Nutzerzahl wird auf etwa 25 Millionen geschätzt.<sup>[3]</sup> Neben Ubuntu selbst, das von Version 11.04 bis 17.04 standardmäßig die von der Ubuntu-Entwicklergemeinschaft selbst entwickelte [Desktop-Umgebung Unity](#) einsetzte und ab Version 17.10 wieder auf [Gnome](#) basiert, existieren verschiedene [Abwandlungen](#). Zu den offiziellen Unterprojekten gehören unter anderem [Kubuntu](#) mit [KDE](#), [Xubuntu](#) mit [Xfce](#), [Ubuntu MATE](#) mit [MATE](#), [Ubuntu Budgie](#) mit [Budgie](#) sowie [Ubuntu Studio](#), das speziell auf die Anforderungen von Audio-, Grafik- und Videobearbeitung ausgerichtet ist. Neue Ubuntu-Versionen erscheinen jedes halbe Jahr im April (04er-Versionen) und im Oktober (10er-Versionen). Die derzeit aktuelle Version 22.04 LTS „Jammy Jellyfish“ erschien am 21. April 2022.

## 1.7.3.2 [ubuntuusers.de](#)

ubuntuusers.de ist ein Deutsches Forum für Endanwender. Es handelt sich um ein sehr umfangreiches Wiki, welches aktiv und aktuell ist.



Thema	Erstellt am	Von
<b>Systemverwaltung, Installation, Aktualisierung</b>		
<b>Vor der Installation und grundlegende Fragen</b>	17. Mai 2022 09:36	von <a href="#">hakei2022</a>
Hier werden Fragen rund um die bevorstehende Installation von Ubuntu beantwortet. Unterforen: <a href="#">Kaufberatung</a>		
<b>Ubuntu installieren und aktualisieren</b>	16. Mai 2022 20:16	von <a href="#">Lidux</a>
Du hast Schwierigkeiten Ubuntu zu installieren oder zu aktualisieren? Dann bist du hier richtig.		
<b>Netzwerk und Internetzugang einrichten</b>	16. Mai 2022 21:22	von <a href="#">Jenswede</a>
Probleme beim Aufbau einer Internetverbindung über DSL-, Analog-Modem, WLAN-, Ethernet-Router, ISDN, UMTS oder sonstige Zugangsverfahren		
<b>System einrichten und verwalten</b>	17. Mai 2022 11:10	von <a href="#">DJKUhpisse</a>
Fragen zur allgemeinen Systemkonfiguration, Benutzerverwaltung etc. Unterforen: <a href="#">Backups</a>		
<b>Grafische Oberfläche</b>		
<b>Grafik, Tastatur und Maus</b>	16. Mai 2022 19:47	von <a href="#">von.wert</a>
Fragen zu Eingabegeräten und der Grafikkonfiguration		
<b>GNOME (Ubuntu ab 17.10)</b>	17. Mai 2022 08:39	von <a href="#">motongroup</a>
Der Standarddesktop von Ubuntu ab 17.10		
<b>Unity (Ubuntu bis 17.04)</b>	20. März 2022 11:30	von <a href="#">beest2106</a>
Der Standarddesktop von Ubuntu bis 17.04		
<b>Ubuntu Touch</b>	8. Mai 2022 15:13	von <a href="#">dino0815</a>
Forum zur Installation und Bedienung der Oberfläche von Ubuntu Touch		
<b>KDE Plasma (Kubuntu)</b>	16. Mai 2022 12:22	

## 1.7.3.3 [askubuntu.com](#)

askubuntu.com ist ein internationales Ubuntu-Forum. Hier bekommt man Informationen zu Bash-Befehlen, Skriptentwicklung und Linux-Internem.

SlackExchange Search on Ask Ubuntu... Log in Sign up

ask ubuntu

Home Explore our Questions Ask Question

PUBLIC Questions Tags Users Companies Unanswered TEAMS Stack Overflow for Teams

boot networking command-line 14.04 16.04 drivers dual-boot 12.04 apt 18.04 more tags Active 8 Bountied Hot Week Month

1 question with new activity

0 votes How can I capitalise a word preceding a specific word 2 answers bash sed awk UnderTheHoud 1 modified 3 mins ago 14 views

0 votes How to Tell Activity Monitor to Measure in GB instead of GiB? 0 answers files process AlexFullinator 962 asked 8 mins ago 5 views

1 vote UFW block by hostname 1 answer ufw hostname Community Bot 1 modified 14 mins ago 3k views

1 vote PHP 7.4 Warning: Unable to load dynamic library 'sqlsrv.so' and 'pdo\_sqlsrv.so' 2 answers server apache2 php sql Vijay A 1 modified 18 mins ago 5k views

0 votes After install doesnt seem to want to boot into desktop 1 answer boot Andreas R 1 answered 25 mins ago 9 views

Hot Network Questions

- Did Russia once want to join NATO? Does former Russian Prime Minister Mikhail Kasanov refer to a serious interest in the past?
- There's more than one way to skin a set
- Bash script segmentation fault
- Toggle 2 LEDs using a timer XMEGA-A3BU Xplained
- "Mach das so." what does mean? How can I understand it?
- What was the first software company to go public?
- Unusually high F value ANOVA
- What is Thomas Aquinas's "Dec. præc."?
- Why is a set with one element distinct from the element itself?
- How to tell my managers manager that I don't want to work under my manager anymore
- How important is to present your paper yourself?
- A small piece of tool fell through the space above the front wheel
- How plausible is it that "a portion of the ocean's floor" could suddenly be "thrown up to the surface"?
- Usefulness of Splendor rare moves
- Are brake calipers directional?
- Lean: dubious noncomputability
- Is it ethical to ask to see a letter of recommendation if the recommender is dead?
- What are the effects of removing P, I, and D from a PID controller respectively?
- Are there exoplanets as large as the star they ...

### 1.7.3.4 linux.org

linux.org ist ein internationales Linux-Forum und bietet allgemeine Linux Hilfe und Tutorials für die Bash.

LINUX.ORG Forums What's new Linux Tutorials Members Download Linux Newsletter Credits Log in Register Search

Latest activity Register

Linux.org

LFCS – Kerberos Authentication (Ubuntu) Stream Music from Linux to Google Home \$100 Digital Ocean Credit Get a free VM to test out Linux!

MAY 11 Kerberos is an authentication method which helps improve security between systems. For systems that need to access another, such as Secure Shell (SSH), Kerberos is a great way to keep the... 1,199 - 0

MAY 03 If anyone has a Google Hub device of some sort, they know how handy they can be for getting information when you ask a question. But there is another use that most people may not be aware of... 1,902 - 3

Bash 05 – Script Logic Live - Let's talk dev-ops - introduction

APR 27 Probably the most important aspect of scripting is using logic operators to control the flow of the script. Sometimes a script needs to do more than run each command in order from the beginning to... 2,589 - 5

APR 22 It was a quick one, but I did a 22ish minute live this thursday with an intro to some great dev-ops / infrastructure tools. Check it out below. I'm going to try and do a weekly live on Thursdays... 2,898 - 2

Bottles - Run Windows software on Linux with Bottles LFCS – Kerberos Authentication (CentOS7)

APR 19 Bottles is a Graphical User Interface (GUI) to aid in setting up the Windows Emulator (WINE). Bottles has many benefits over WINE by itself. 4,609 - 1

APR 11 Kerberos is an authentication method that can assign a user a 'ticket' after the first sign-on. Once a user has entered a correct password, then they are granted a 'ticket' to allow connection... 3,564 - 0

Members online

Chris-0001, kctldi, jifulinx, Bobbykic

Total: 445 (members: 6, guests: 439)

Latest posts

- Camera not working Latest: jifulinx - 9 minutes ago Linux Audio / Video
- Are we suffering attempted DDoS attacks. Latest: Brickwizard - 55 minutes ago Forum Assistance
- DMB Digital Media Broadcast Latest: magmas - 58 minutes ago Linux Networking
- Wrong location in Redshift Latest: AnonymousPigeon - Today at 11:44 AM Getting Started
- Disk select Latest: Teja Chabbi - Today at 11:15 AM General Linux

For anyone who uses WINE, you need to look into...











Forums > [Linux Tutorials](#) >

## Linux Beginner Tutorials

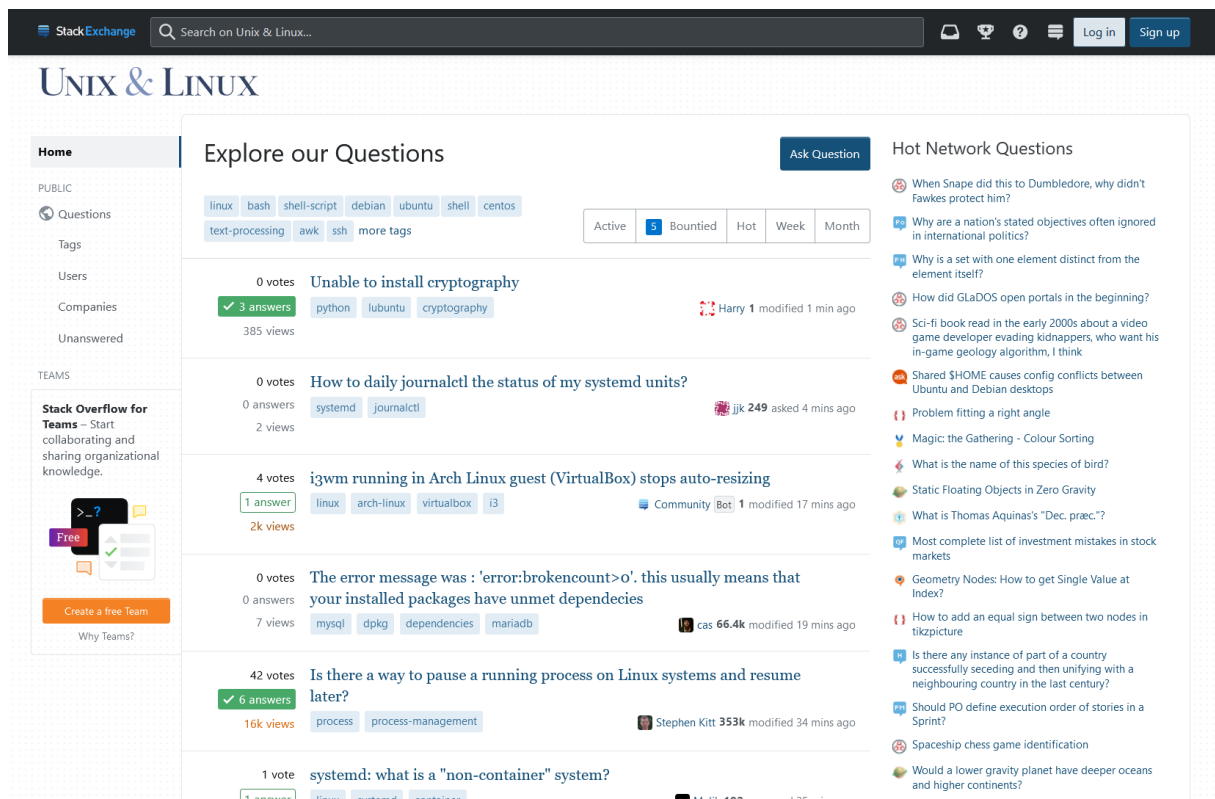
1 2 Next >

Filters ▾

	<b>Introduction to Linux</b> Rob · Jul 2, 2013 <a href="#">6</a> <a href="#">7</a> <a href="#">8</a>	Replies: <b>140</b> Views: 440K	Apr 7, 2022 the hacker dream 
	<b>Clear! (clear your terminal screen)</b> Rob · Apr 25, 2017	Replies: <b>17</b> Views: 152K	Dec 19, 2021 SpongebobFan1994 
	<b>More useful commands</b> Rob · Jul 9, 2013	Replies: <b>10</b> Views: 138K	Nov 17, 2021 alvaro56 
	<b>File Permissions - chmod</b> Rob · Jul 9, 2013 <a href="#">2</a>	Replies: <b>31</b> Views: 2M	Oct 19, 2021 mrp 

### 1.7.3.5 [unix.stackexchange.com](https://unix.stackexchange.com)

unix.stackexchange.com ist ein Forum für Programmierer. Es gibt Informationen zu Bash-Befehlen, der Skriptentwicklung und Linux-Internem.



The screenshot shows the Stack Exchange website interface for the 'UNIX & LINUX' tag. The top navigation bar includes the Stack Exchange logo, a search bar, and user options like 'Log in' and 'Sign up'. The main content area is titled 'Explore our Questions' and features a list of questions with filters for 'Active', 'Bountied', 'Hot', 'Week', and 'Month'. The questions listed include:

- Unable to install cryptography**: 0 votes, 3 answers, 385 views. Tags: python, ubuntu, cryptography. Modified 1 min ago by Harry.
- How to daily journal the status of my systemd units?**: 0 votes, 0 answers, 2 views. Tags: systemd, journalctl. Asked 4 mins ago by jjk.
- i3wm running in Arch Linux guest (VirtualBox) stops auto-resizing**: 4 votes, 1 answer, 2k views. Tags: linux, arch-linux, virtualbox, i3. Modified 17 mins ago by Community Bot.
- The error message was : 'error:brokencount>o'. this usually means that your installed packages have unmet dependencies**: 0 votes, 0 answers, 7 views. Tags: mysql, dpkg, dependencies, mariadb. Modified 19 mins ago by cas.
- Is there a way to pause a running process on Linux systems and resume later?**: 42 votes, 6 answers, 16k views. Tags: process, process-management. Modified 34 mins ago by Stephen Kitt.
- systemd: what is a "non-container" system?**: 1 vote, 1 answer. Tags: linux, systemd, container. Answered 35 mins ago by Malik.

On the right side, there is a 'Hot Network Questions' section with a list of trending questions, such as 'When Snape did this to Dumbledore, why didn't Fawkes protect him?' and 'Why are a nation's stated objectives often ignored in international politics?'.

## Rename files with numbers already in file name

Asked today Modified today Viewed 18 times

I have files with following names starting with water-frames0.gro upto water-frames201.gro

```
1 water-frames0.gro water-frames119.gro water-frames138.gro water-frames157.gro
water-frames116.gro water-frames135.gro water-frames154.gro
```

How do I add leading zeros and dash before numbers so that the files are sorted correctly on the terminal? I need this to work with thousands of file names, so I imagine adding a few extra zeros is useful.

I imagine the new filenames to be something as

```
water-frames-0000.gro water-frames-0119.gro water-frames-0138.gro water-frames-0157.gro
water-frames-0116.gro water-frames-0135.gro water-frames-0135.gro water-frames-0154.gro
```

I tried playing with rename and looking at previous questions, however, I couldn't find something that I could adapt.

Thank you,

shell-script terminal rename mmv

### 1.7.3.6 stackoverflow.com

Stackoverflow ist das größte Forum für Programmierer. Man erhält Informationen zu Bash-Befehlen und der Skriptentwicklung, aber manchmal herrscht ein rauher Ton.

The screenshot shows the Stack Overflow website interface. At the top, there is a navigation bar with the Stack Overflow logo, links for 'About', 'Products', and 'For Teams', a search bar containing 'bash mass rename', and buttons for 'Log in' and 'Sign up'. Below the navigation bar, a banner for the '2022 Developer Survey' is visible. The main content area is divided into three columns. The left column contains navigation links for 'Home', 'PUBLIC' (Questions, Tags, Users, Companies), 'COLLECTIVES' (Explore Collectives), and 'TEAMS' (Stack Overflow for Teams). The middle column displays 'Search Results' for 'bash mass rename', showing 24 results. The top result is 'Mass rename files (bash)' with 1 answer and 985 views. The second result is 'Mass rename folders and their files in bash/terminal' with 2 answers and 588 views. The third result is 'Bash script to mass rename mp3s? [closed]' with 1 answer and 103 views. The fourth result is 'bash mass rename files in folders and subfolders with sequential number' with 2 answers and 685 views. The fifth result is 'bash - mass renaming files with many special characters' with 2 answers and 2k views. The right column is titled 'Hot Network Questions' and lists several trending questions, such as 'How can I deal with students that are too sensitive when they are simply wrong?' and 'What is Thomas Aquinas's "Dec. præc.?"

2 Answers Sorted by: Highest score (default) ▾

- EDIT: Better approach with a function rather than recursive calls to the same script.
- 2 Here's hoping I got all corner cases. This descends into directories recursively to handle deeply nested directory trees.
- ✓
 Caveat: Because the script takes care to not overwrite existing files, gaps may appear in the numbering in some corner cases -- if there is a file `0.txt` in a directory and the first file that is handled in that directory is a `.txt` file, it will be moved to `1.txt`. This also happens if the first file that is handled is `0.txt`, so running the script twice will change the numbering, and running it again will change it back.

So here's the code:

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

### 1.7.3.7 Lösung suchen

Im Englischen findet man meist mehr Lösungen. Um Hilfe zu bekommen, sollte man seine Fehlermeldung kopieren oder das Problem anderweitig beschreiben. Man kann auch versuchen, bei der Beschreibung Synonyme zu verwenden und sollte eine möglichst einfache Sprache verwenden. Jedoch sollte die Sprache spezifisch genug sein, um sein Problem von Ähnlichen abzugrenzen. Weiterhin kann man versuchen, nach der Hauptdistribution zu suchen. Beispielsweise wenn man Linux Mint hat, sucht man nach Ubuntu, oder wenn man Kali hat, sucht man nach Debian. Außerdem sollte man versuchen, das Problem weiter einzugrenzen bzw. zu verstehen. Geht das Internet nicht, kann man schauen, ob die IP-Adresse richtig konfiguriert wurde, ob das Gateway erreichbar ist, ob eine externe Adresse erreichbar ist, ob der DNS funktioniert (nslookup), oder Ähnliches.

### 1.7.3.8 Problemlösung im Forum

Beginnend sollte man sich der Problembeschreibung widmen, sprich was möchte ich erreichen, was habe ich bereits probiert, woran scheitere ich gerade und wurde das Problem schon einmal beschrieben? Wenn ja sollte man schauen, was dort nicht funktioniert. Dann sollte man schauen, welche Umgebung man benutzt, sprich welches Betriebssystem, welche Programmversion, aktuelle Ausgabe und detaillierte Infos zur Hardware. Zusätzlich kann man schauen, wie sich das Problem replizieren lässt.

Um die vorgeschlagenen Lösungen zu testen, sollte man beginnend ein Backup anlegen. Außerdem sollte man versuchen die Antwort zu verstehen, anstatt sie blind zu kopieren. Wenn die Lösung nicht erfolgreich war, kann man das erstellte Backup wieder einspielen und die Einstellungen rückgängig machen. Wenn man eine funktionierende Lösung gefunden hat, sollte man eine eigene Beschreibung erstellen, was funktioniert hat. Somit hilft man anderen Nutzern mit demselben Problem.

## 1.7.4 Blogs und Webseiten

### 1.7.4.1 [problem-hilfe.de/linux](http://problem-hilfe.de/linux)

Auf [problem-hilfe.de/linux](http://problem-hilfe.de/linux) findet man deutsche Erklärungen. Es gibt Informationen zu Linux allgemein, zur Bash, zu Skripten, zu Oberflächen, zu Server-Programmen und zu Konfigurationen.

## Grundlagen von Linux

\* Einführung \* Installation \* Paketmanager \* Grundlagen \* Shells \* Entwicklung \* KDE \* Skriptsprachen \* AWK \* PHP \* Perl \* Apache \* Veranstaltungen \* Schulungen \* Bücher \* Netzwerk \* OpenOffice \* OpenSource \* Samba

### Grundlagen



Home

- Benutzerverwaltung
- Dateiverwaltung
- Root-Passwort vergessen
- Verzeichnisstruktur
- Zugriffsrechte im Dateisystem

Mit freundlicher Unterstützung von:

Linux-Kurse und Seminare



### Benutzerverwaltung

"Treffen Einfach und Gründlichkeit zusammen, entsteht Verwaltung". Oliver Hassencamp

Linux ist ein Multiuser-System und benötigt Mechanismen, die festlegen, wer auf welche Dateien und Verzeichnisse zugreifen darf und wer welche Programme starten darf. Dazu gibt es unter Linux Benutzer und Gruppen. Jeder Benutzer ist mindestens einer Gruppe zugeordnet. Für jeder Datei (und damit auch jedem Programm) ist ein Besitzer, eine Gruppenzugehörigkeit und ein Zugriffsbit gespeichert.

Es gibt drei Arten von Benutzer:

1. Der Super-User mit dem Namen „root“ hat uneingeschränkte Rechte, d.h. root darf alle Dateien ansehen, verändern, löschen, alle Programme starten usw.
2. gewöhnliche Benutzer haben uneingeschränkten Zugriff auf ihre eigenen Dateien aber nur eingeschränkten Zugriff auf den Rest des Dateisystems.
3. Benutzer für Dämonen und Server-Dienste, die nicht für den interaktiven Einsatz vorgesehen sind, z.B. „httpd“ als Benutzer für den Webserver Apache.

Benutzernamen:

Der Benutzer-Name (Login-Name) sollte nur aus Kleinbuchstaben bestehen und nach Möglichkeit nicht länger als acht Zeichen sein, denn sonst könnte es Probleme mit bestimmten Programmen geben. Beispielsweise unterscheidet FTP nicht zwischen Groß- und Kleinschreibung.



Aufbau von passwd

Die Datei /etc/passwd enthält eine Liste aller Benutzer

Beispiel: Auszug aus einer passwd-Datei:

```
atx:25:25:Batch jobs daemon:/var/spool/atjobs:/bin/bash
beagleindex:x:108:110:User for Beagle indexing:/var/cache/beagle:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
daemon:x:2:2:Daemon:/sbin:/bin/bash
dhcpcd:x:102:65534:DHCP server daemon:/var/lib/dhcp:/bin/false
ftp:x:40:49:FTP account:/srv/ftp:/bin/bash
....
```

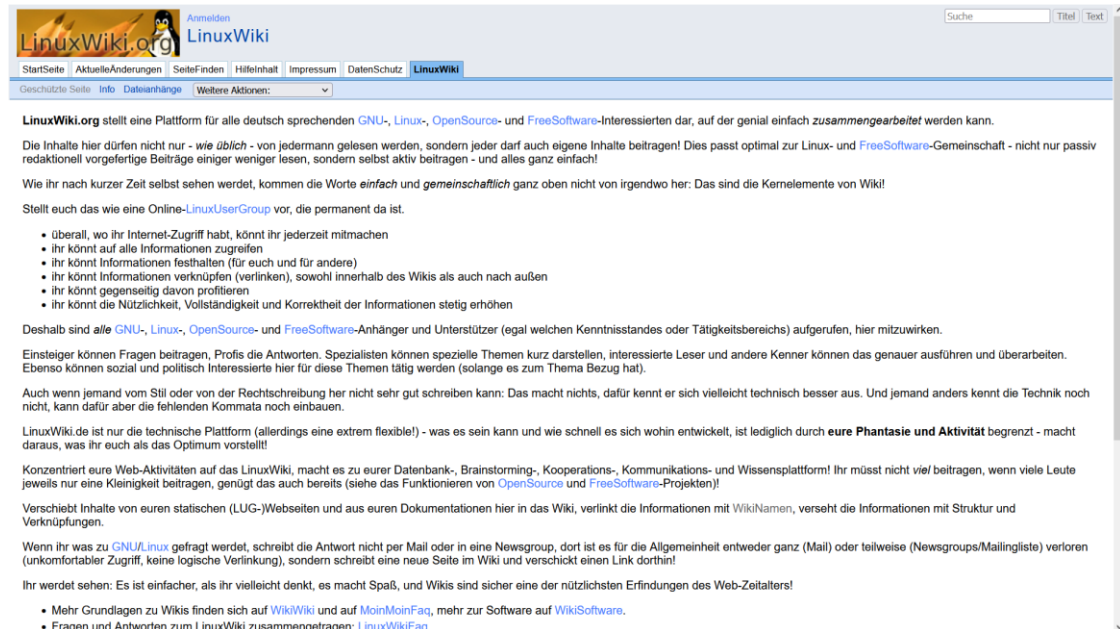
### 1.7.4.2 Thomas Krenn Wiki

Bei Thomas Krenn Wiki gibt es deutsche Erklärungen zu Linux, Hardware-Berichte, Praxiserfahrungen, Infos zum Netzwerk und zur Serveradministration. Die Erklärungen sind sehr verständlich dargelegt.

The screenshot shows the Thomas Krenn Wiki interface. At the top, there is a search bar with the text 'Thomas-Krenn-Wiki durchsuchen' and a search icon. To the right of the search bar are buttons for 'Onlineshop' and 'DE -'. Below the search bar is a navigation bar with the 'wiki' logo, 'Lesen', 'Quelltext', and 'Versionsgeschichte' options. On the right side of the navigation bar, there are links for 'Benutzerkonto erstellen' and 'Anmelden'. The main content area is titled 'OpenVPN Grundlagen'. Below the title is a breadcrumb trail: 'Hauptseite > Server-Software > Linux'. The main text describes OpenVPN as free software for setting up Virtual Private Networks (VPN) over encrypted TLS/SSL connections. It mentions that the transport protocol can be UDP or TCP, and authentication methods include Pre-shared Key and certificates. A reference is provided: 'OpenVPN ist unter der GNU GPL (General Public License) lizenziert.<sup>[1]</sup>'. Below the text is a table of contents with the following items: 1 Eigenschaften, 2 Vorteile, 3 Authentifizierung (with sub-items 3.1 Pre-shared Key and 3.2 Zertifikatsbasiert), 4 Kommunikation (with sub-items 4.1 Netzwerkmodi, 4.1.1 Bridging (TAP-Device), and 4.1.2 Routing (TUN-Device)), and 5 Einzelnachweise. Below the table of contents is a section titled 'Eigenschaften' which lists the following features of OpenVPN: tunneling any subnetwork or virtual Ethernet adapters, scalability, configuration flexibility, use of OpenSSL library, authentication methods, bridging and routing modes, dynamic endpoints, and GUI control.

### 1.7.4.3 linuxwiki.de

linuxwiki beinhaltet deutsche Erklärungen zu Linux, zu Anwendungsfällen, zu Beispielen und zu Erfahrungsberichten. Jedoch ist die Navigation sehr unübersichtlich.



**LinuxWiki.org** stellt eine Plattform für alle deutsch sprechenden GNU-, Linux-, OpenSource- und FreeSoftware-Interessierten dar, auf der genial einfach *zusammengearbeitet* werden kann.

Die Inhalte hier dürfen nicht nur - wie üblich - von jedermann gelesen werden, sondern jeder darf auch eigene Inhalte beitragen! Dies passt optimal zur Linux- und FreeSoftware-Gemeinschaft - nicht nur passiv redaktionell vorgefertigte Beiträge einiger weniger lesen, sondern selbst aktiv beitragen - und alles ganz einfach!

Wie ihr nach kurzer Zeit selbst sehen werdet, kommen die Worte *einfach* und *gemeinschaftlich* ganz oben nicht von irgendwo her: Das sind die Kernelemente von Wiki!

Stellt euch das wie eine Online-**LinuxUserGroup** vor, die permanent da ist.

- überall, wo ihr Internet-Zugriff habt, könnt ihr jederzeit mitmachen
- ihr könnt auf alle Informationen zugreifen
- ihr könnt Informationen festhalten (für euch und für andere)
- ihr könnt Informationen verknüpfen (verlinken), sowohl innerhalb des Wikis als auch nach außen
- ihr könnt gegenseitig davon profitieren
- ihr könnt die Nützlichkeit, Vollständigkeit und Korrektheit der Informationen stetig erhöhen

Deshalb sind *alle* GNU-, Linux-, OpenSource- und FreeSoftware-Anhänger und Unterstützer (egal welchen Kenntnisstandes oder Tätigkeitsbereichs) aufgerufen, hier mitzuwirken.

Einsteiger können Fragen beitragen, Profis die Antworten. Spezialisten können spezielle Themen kurz darstellen, interessierte Leser und andere Kenner können das genauer ausführen und überarbeiten. Ebenso können sozial und politisch Interessierte hier für diese Themen tätig werden (solange es zum Thema Bezug hat).

Auch wenn jemand vom Stil oder von der Rechtschreibung her nicht sehr gut schreiben kann: Das macht nichts, dafür kennt er sich vielleicht technisch besser aus. Und jemand anders kennt die Technik noch nicht, kann dafür aber die fehlenden Kommata noch einbauen.

LinuxWiki.de ist nur die technische Plattform (allerdings eine extrem flexible!) - was es sein kann und wie schnell es sich wohin entwickelt, ist lediglich durch **eure Phantasie und Aktivität** begrenzt - macht daraus, was ihr euch als das Optimum vorstellt!

Konzentriert eure Web-Aktivitäten auf das LinuxWiki, macht es zu eurer Datenbank-, Brainstorming-, Kooperations-, Kommunikations- und Wissensplattform! Ihr müsst nicht viel beitragen, wenn viele Leute jeweils nur eine Kleinigkeit beitragen, genügt das auch bereits (siehe das Funktionieren von OpenSource und FreeSoftware-Projekten)!

Verschiebt Inhalte von euren statischen (LUG-)Webseiten und aus euren Dokumentationen hier in das Wiki, verlinkt die Informationen mit WikiNamen, verseht die Informationen mit Struktur und Verknüpfungen.

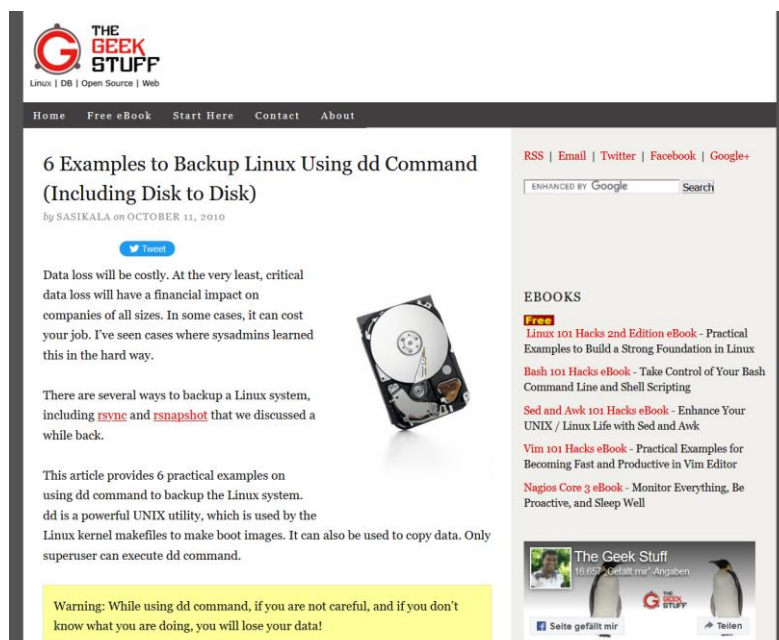
Wenn ihr was zu GNU/Linux gefragt werdet, schreibt die Antwort nicht per Mail oder in eine Newsgroup, dort ist es für die Allgemeinheit entweder ganz (Mail) oder teilweise (Newsgroups/Mailingliste) verloren (unkomfortabler Zugriff, keine logische Verlinkung), sondern schreibt eine neue Seite im Wiki und verschickt einen Link dorthin!

Ihr werdet sehen: Es ist einfacher, als ihr vielleicht denkt, es macht Spaß, und Wikis sind sicher eine der nützlichsten Erfindungen des Web-Zeitalters!

- Mehr Grundlagen zu Wikis finden sich auf [WikiWiki](#) und auf [MoinMoinFaq](#), mehr zur Software auf [WikiSoftware](#).
- Fragen und Antworten zum LinuxWiki zusammengetragen: [LinuxWikiFaq](#)

### 1.7.4.4 thegeekstuff.com

thegeekstuff ist ein englischer Blog. Man findet dort Erfahrungsberichte, Blogbeiträge zu speziellen Themen, Befehle, welche an Beispielen erklärt werden und Links für Bücher.



**THE GEEK STUFF**  
Linux | DB | Open Source | Web

Home | Free eBook | Start Here | Contact | About

## 6 Examples to Backup Linux Using dd Command (Including Disk to Disk)

by SASIKALA on OCTOBER 11, 2010

[Tweet](#)

Data loss will be costly. At the very least, critical data loss will have a financial impact on companies of all sizes. In some cases, it can cost your job. I've seen cases where sysadmins learned this in the hard way.

There are several ways to backup a Linux system, including [rsync](#) and [rsnapshot](#) that we discussed a while back.

This article provides 6 practical examples on using dd command to backup the Linux system. dd is a powerful UNIX utility, which is used by the Linux kernel makefiles to make boot images. It can also be used to copy data. Only superuser can execute dd command.

**Warning: While using dd command, if you are not careful, and if you don't know what you are doing, you will lose your data!**

RSS | Email | Twitter | Facebook | Google+

ENHANCED BY Google  Search

**EBOOKS**

- Free** [Linux 101 Hacks 2nd Edition eBook](#) - Practical Examples to Build a Strong Foundation in Linux
- [Bash 101 Hacks eBook](#) - Take Control of Your Bash Command Line and Shell Scripting
- [Sed and Awk 101 Hacks eBook](#) - Enhance Your UNIX / Linux Life with Sed and Awk
- [Vim 101 Hacks eBook](#) - Practical Examples for Becoming Fast and Productive in Vim Editor
- [Nagios Core 3 eBook](#) - Monitor Everything, Be Proactive, and Sleep Well

The Geek Stuff  
16.637 "Gefällt mir"-Angaben

Seite gefällt mir [Teilen](#)

### 1.7.4.5 elektronik-kompedium.de

elektronik-kompedium ist ein deutscher Blog, welcher Erklärungen zur Hardware liefert. Netzwerktechnologien wie DHCP, IPv6, NAT oder Ähnliches werden ausführlich erläutert. elektronik-kompedium liefert eher allgemeine Informationen, da der Fokus nicht unbedingt auf Linux liegt.

### 1.7.4.6 geeksforgeeks.org

geeksforgeeks bietet ein breites Angebot. Dazu gehören Tutorials für Programmierer, Erklärungen zu Betriebssystemen und Beispiele für Linux. Die Erklärungen sind generell sehr ausführlich.

The screenshot shows the GeeksforGeeks website. The main article is titled "RAID (Redundant Arrays of Independent Disks)" with a difficulty level of "Easy" and a last update date of "22 Sep, 2021". The article text explains that RAID is a technique for using multiple disks for performance and redundancy. It also includes a section on "Why data redundancy?" and "Key evaluation points for a RAID System" with bullet points for Reliability, Availability, and Performance. On the right side, there is a "WHAT'S NEW" section with three featured articles: "Interview Series- Prepare, Practice & Upskill", "Data Structures & Algorithms- Self Paced Course", and "Complete Interview Preparation". Below this is a "MOST POPULAR IN DBMS" section with a link to "SQL | Join (Inner, Left, Right and Full Joins)".

### 1.7.5 Zusammenfassung

Sie kennen nun Möglichkeiten die offiziellen Dokumentationen von Programmen aufzufinden und diese zu verstehen.

Heute haben Sie weitere nützliche Foren kennengelernt, bei denen Sie Hilfe suchen können.

Sie haben in der heutigen Vorlesung eine kleine Übersicht von Webseiten präsentiert bekommen, bei denen Sie Informationen und Tipps zu Bash und Linux erhalten.

Es gibt keine universelle Hilfeseite. Daher sollte Ihnen klar geworden sein, welche Hilfe Sie weiterbringt, wenn Sie ein Problem spezielles Problem haben.

Linux hindert Sie nicht daran dumme Sachen zu tun. Achten Sie also auf Trolle in Lösungen.



## 1.8 Namespaces und Git

### 1.8.1 Git

Git ist eine Software für die Versionsverwaltung von Dateien, wurde von Linux Torvald entwickelt und ist Open Source (GPLv2). Es gibt verschiedene Gründe, warum man Git nutzen sollte. Dazu zählt, dass Skripte eine Zeitersparnis aufgrund der damit erzielten Automatisierung bieten. Git bietet eine solche Zeitersparnis durch den Austausch von Skripten. Es bestehen Unterschiede zwischen den einzelnen Programmversionen und es ist möglich, eine spezielle Version auszuwählen. In Git arbeiten mehrere Nutzer zusammen. Weiterhin ist es zeilenbasiert. Die folgende Abbildung zeigt das Git Logo:



#### 1.8.1.1 Git lokales Arbeiten

Mit Git kann man lokal arbeiten. Um ein Git-Repository lokal anzulegen, kann man entweder ein Verzeichnis, welches nicht unter Versionskontrolle steht, in ein Git-Repository umwandeln, oder man klonet ein bestehendes Git-Repository.

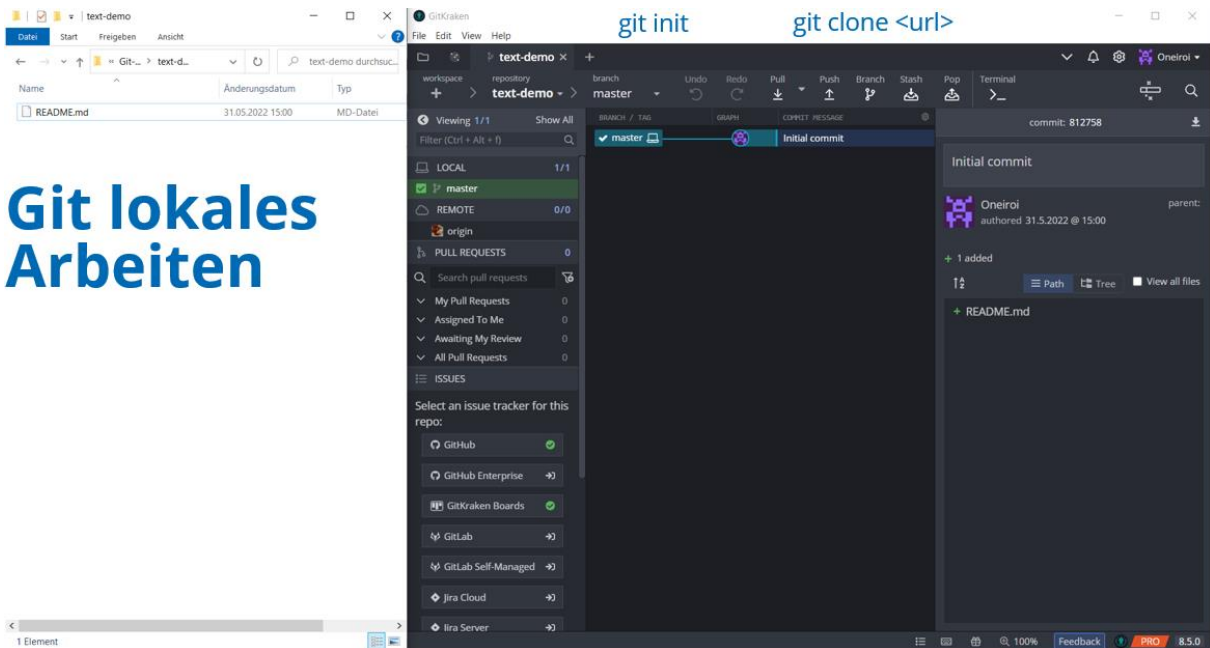
Um ein Verzeichnis ohne Versionskontrolle in ein Git-Repository umzuwandeln, wird zunächst in das Verzeichnis gewechselt. Dort führt man den Befehl „git init“ aus, wodurch ein Unterverzeichnis .git mit allen relevanten Repository-Daten erzeugt wird.

Wird ein existierendes Repository geklont, wird eine Kopie davon angelegt. Dabei werden alle Daten zu dem Projekt, welche auf dem Server liegen, auf den lokalen Rechner. Hierfür wird der Befehl „git clone <url>“ verwendet. Daraufhin wird ein Verzeichnis mit dem Namen des Repositorys angelegt, welches mit einem .git Verzeichnis initialisiert wird. Dabei werden alle Daten des Repositorys heruntergeladen.

In diesem Beispiel wird das Repository text-demo lokal angelegt, in welchem sich die Datei README.md befindet. Die momentane Branch ist die Branch „master“.

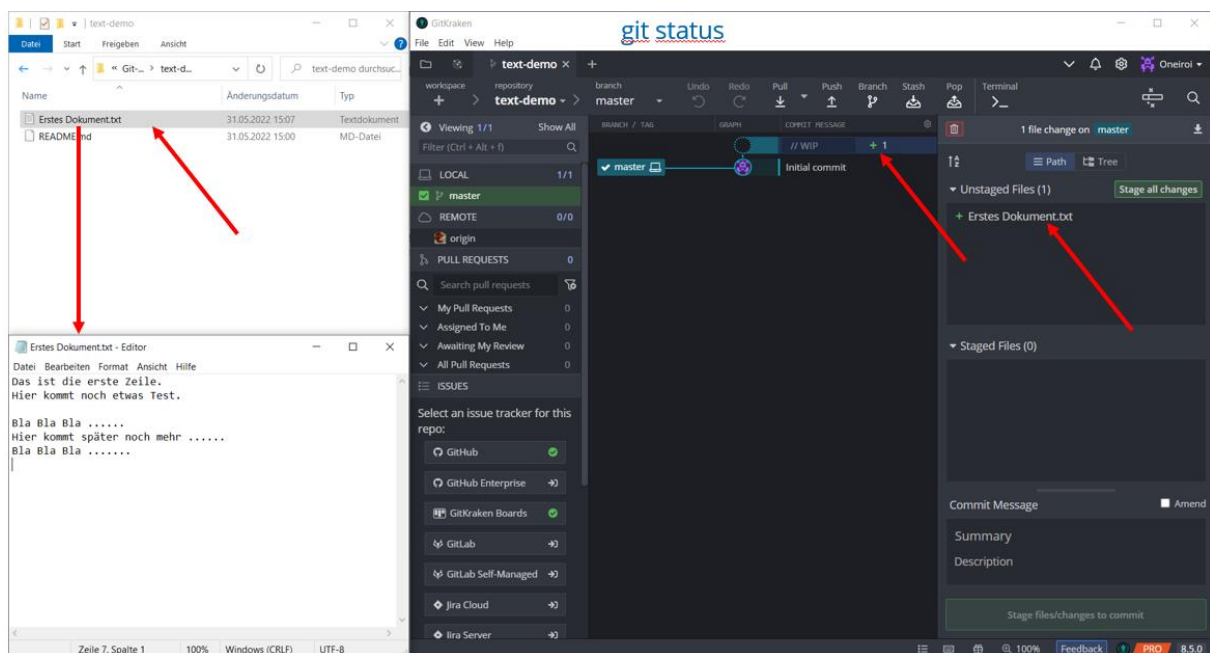


# Git lokales Arbeiten

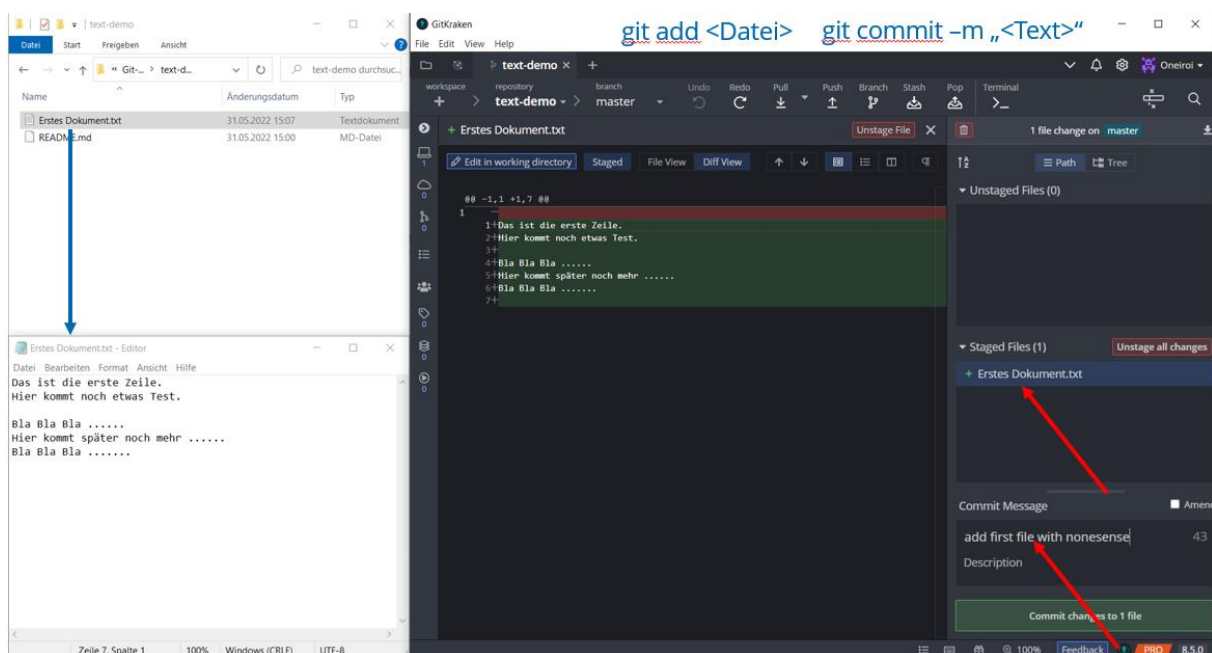


Eine Datei im Repository kann als untracked oder tracked angesehen werden. Der Zustand tracked beschreibt, dass Änderungen an der Datei verfolgt werden. Mit untracked geschieht dies nicht. Tracked Dateien können unverändert, modifiziert oder für den nächsten Commit vorgemerkt (staged) sein.

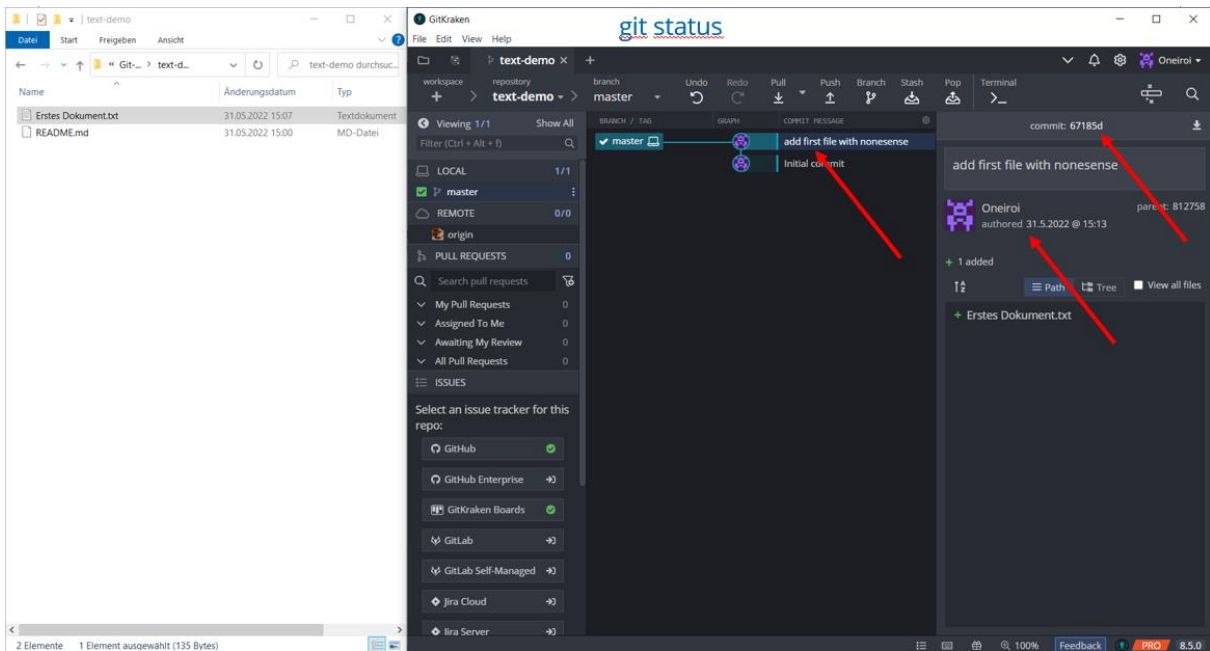
Um diesen Zustand einer Datei zu prüfen, wird der Befehl „git status“ verwendet. In diesem Beispiel wird die Datei „Erstes Dokument.txt“ lokal angelegt. Wird nun der Status der Dateien überprüft, wird diese als „unstaged“ aufgelistet, da sie noch nicht geadded wurde und somit noch nicht versioniert wird.



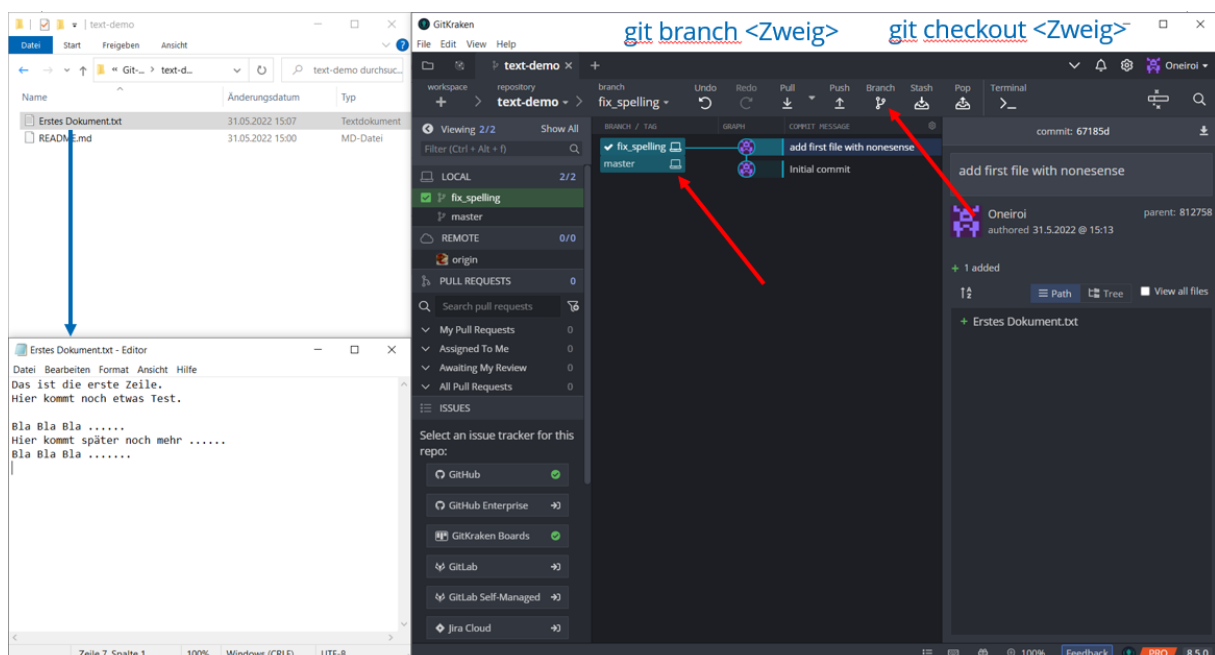
Um zu erreichen, dass für die Datei „Erstes Dokument.txt“ eine Versionsverwaltung durchgeführt wird, wird der Befehl „git add <Datei>“ verwendet. Dieser kann genutzt werden, um Dateien zur Versionsverwaltung hinzuzufügen oder Dateien für einen Commit vorzumerken. Somit würde in diesem Fall „git add Erstes Dokument.txt“ ausgeführt werden. Nun wird die Datei auf den Zustand „staged“ gesetzt, sodass sie für den nächsten Commit vorgemerkt ist. Um die Datei zu committen, wird nun der Befehl „git commit“ ausgeführt. Somit werden alle Dateien, die für einen commit vorgemerkt sind, committed. Soll eine individuelle Commit-Nachricht mit angegeben werden, wird der Befehl „git commit -m „<Text>““ verwendet. In diesem Fall wird die Commit Message „add first file with nonsense“ angegeben und die Datei „Erstes Dokument.txt“ in die Branch „master“ committed.



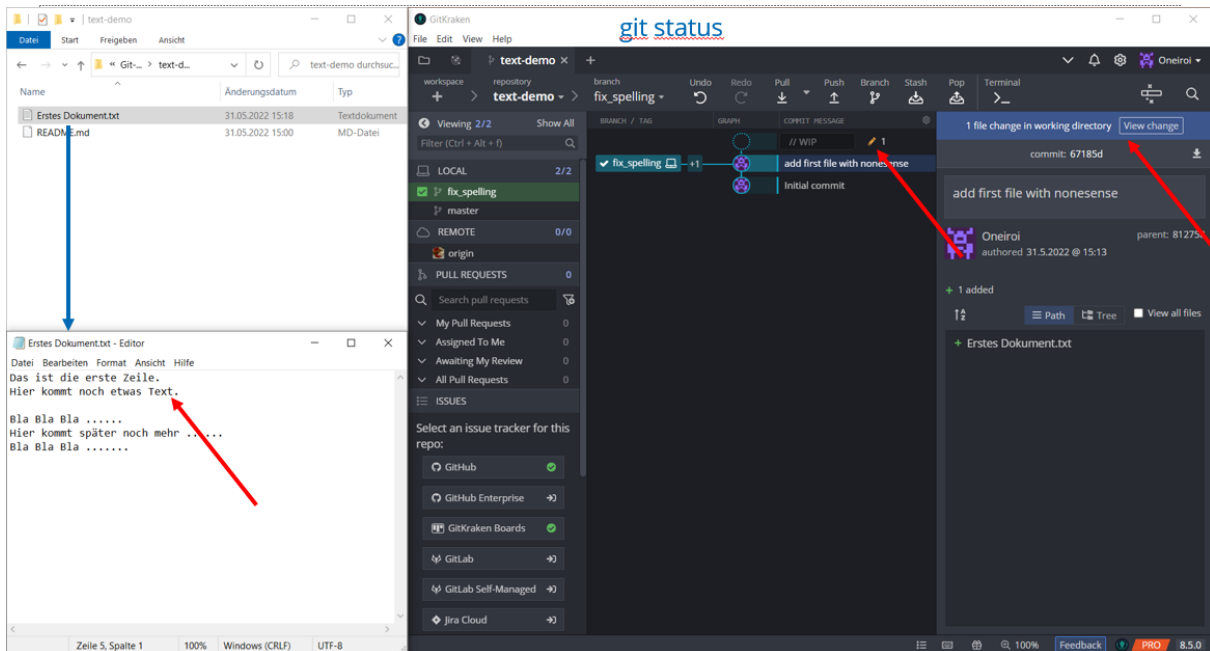
Gibt man sich nun nochmals mit git status den Zustand der Dateien aus sieht man, dass das Dokument mit der Commit Nachricht „add first file with nonsense“ committed wurde. Der Commit hat eine SHA-1-Prüfsumme, welche rechts angegeben ist. In diesem Fall lautet sie „67185d“.



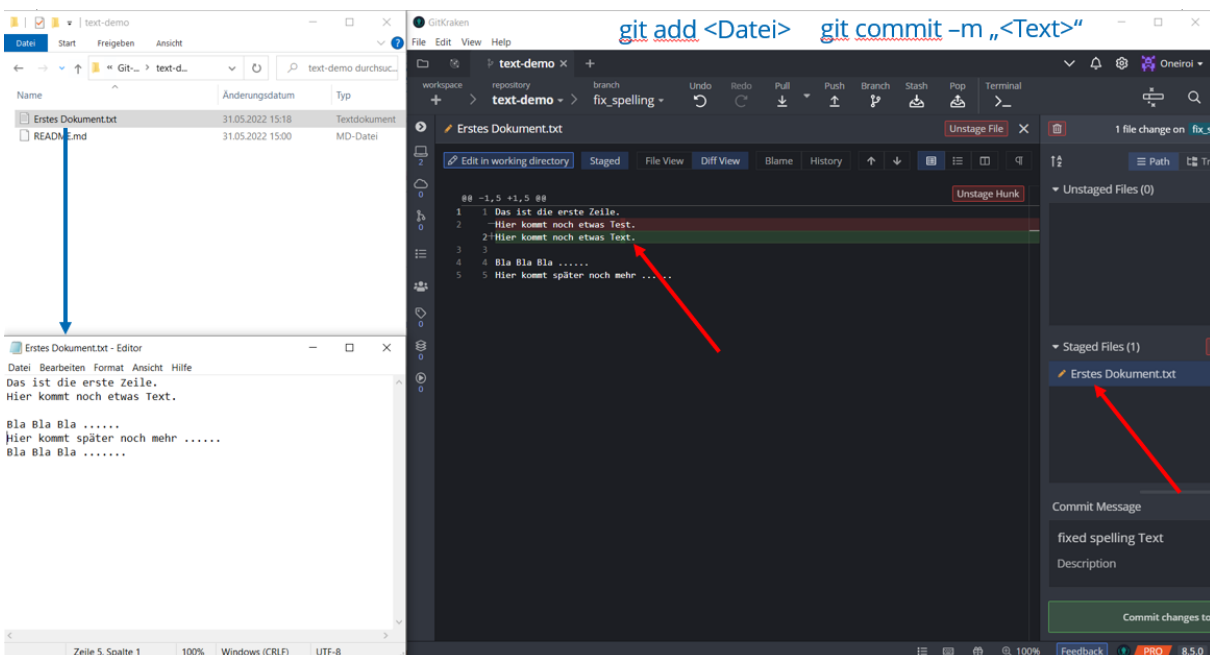
Die Standardbezeichnung einer Branch in Git ist „master“. Diese zeigt auf den letzten Commit, der durchgeführt wurde. In diesem Fall „add first file with nonsense“. Wird eine neue Branch angelegt, wird ein neuer Zeiger erstellt. Dafür wird der Befehl „git branch <Zweig>“ verwendet, in diesem Fall „git branch fix\_spelling“. Dieser neu erzeugte Zeiger zeigt auf denselben Commit, auf welchem man sich gegenwärtig befindet. In diesem Fall zeigt er auf „add first file with nonsense“. Momentan befindet man sich noch in der Branch „master“. Um nun in die erzeugte Branch „fix\_spelling“ zu wechseln, wird der Befehl „git checkout fix\_spelling“ genutzt.



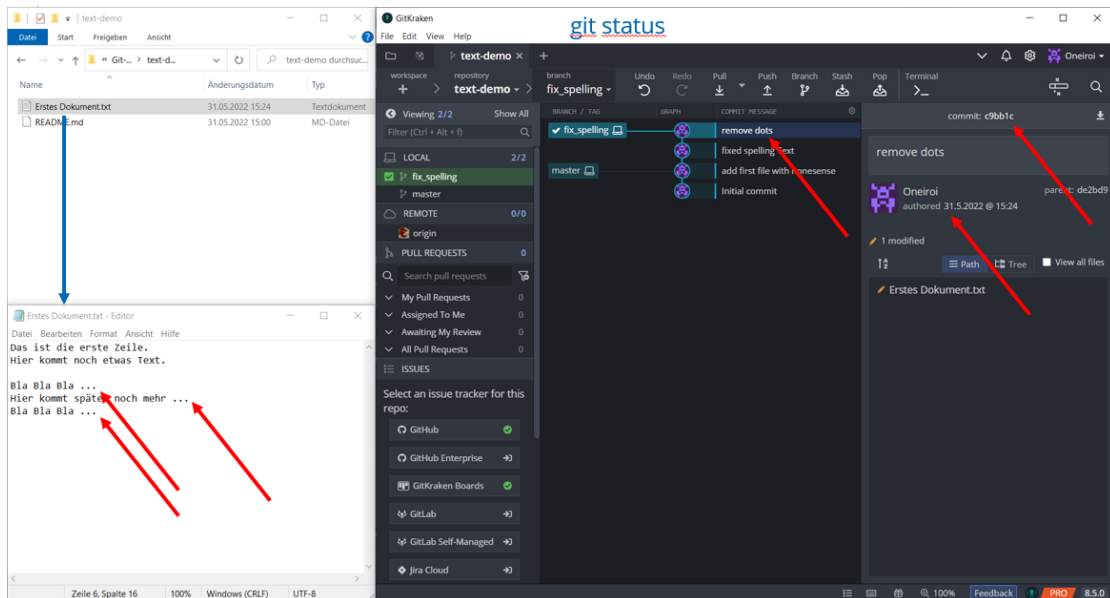
Wird nun etwas in der Datei „Erstes Dokument.txt“ geändert, wird dies mit dem „Stift“-Symbol angezeigt. Über den Befehl „git status“ lässt sich wieder der Zustand der Dateien anzeigen.



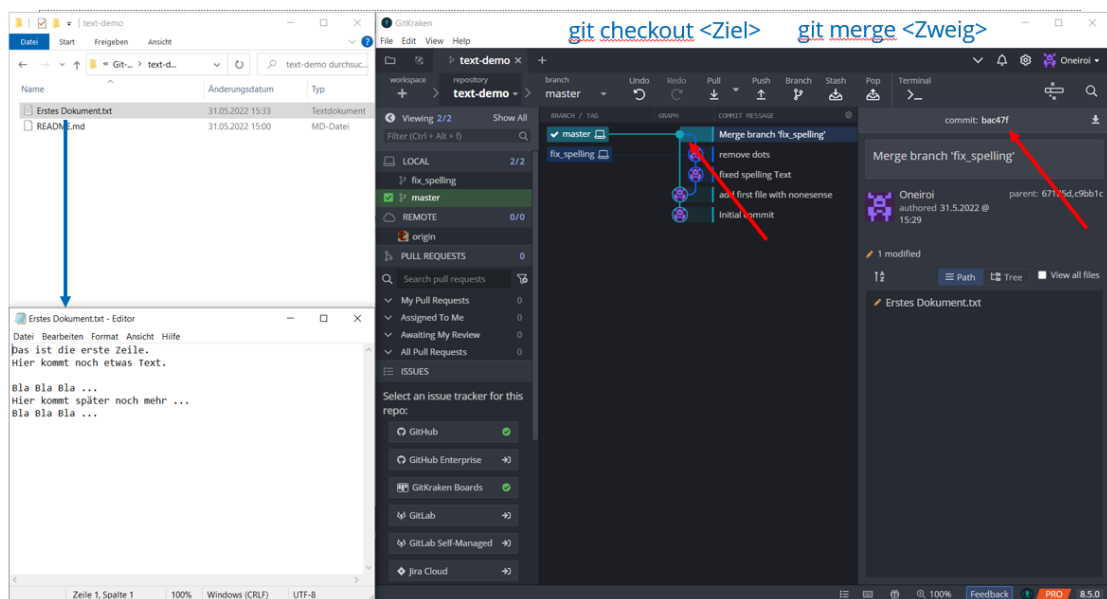
Nun kann die geänderte Datei wieder mit „git add Erstes Dokument.txt“ auf den Zustand „staged“ gesetzt werden, sodass sie für den nächsten Commit vorgemerkt ist. Mit „commit -m „fixed spelling Text“ wird die Datei mit den durchgeführten Änderungen in der aktuellen Branch (fix\_spelling) committed.



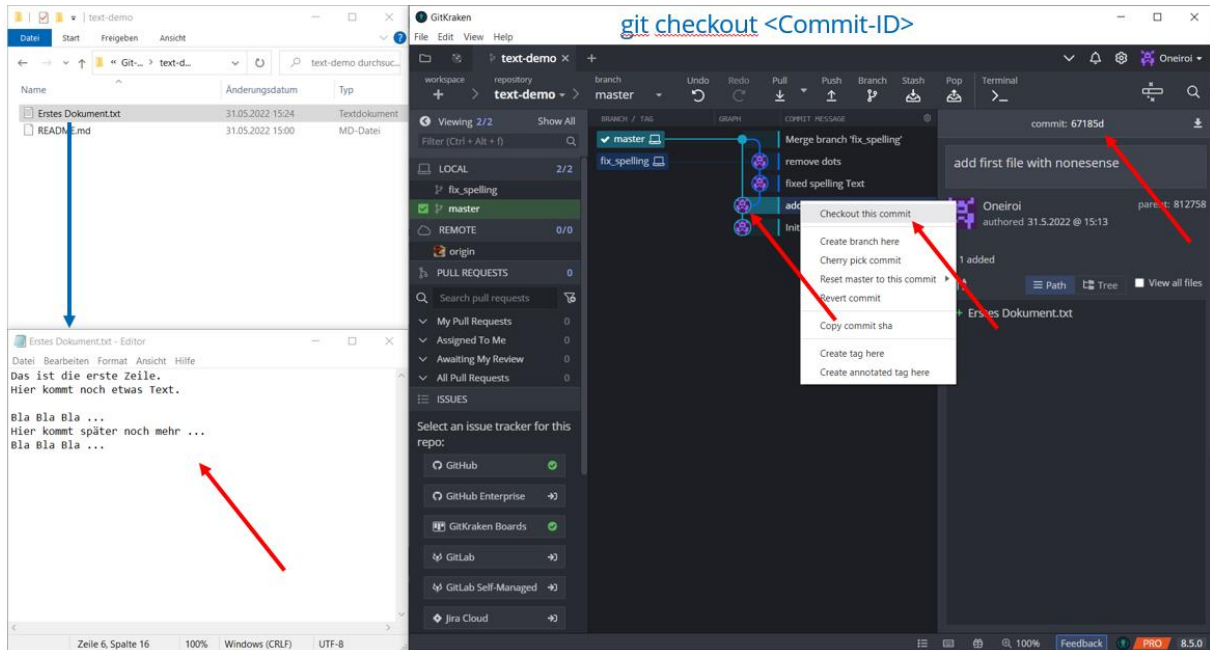
Im nächsten Schritt werden in der Datei „Erstes Dokument.txt“ einige Punkte entfernt. Dieses Dokument wird ebenfalls geaddet und mit dem Text „remove dots“ commited. Somit zeigt die Branch „fix\_spelling“ auf den letzten Commit „remove dots“. Die SH1-Prüfsumme dieses Commits lautet „c9bb1c“. Mit „git status“ lässt sich wieder der Zustand der Datei anzeigen, wobei er in diesem Fall „modified“ lautet.



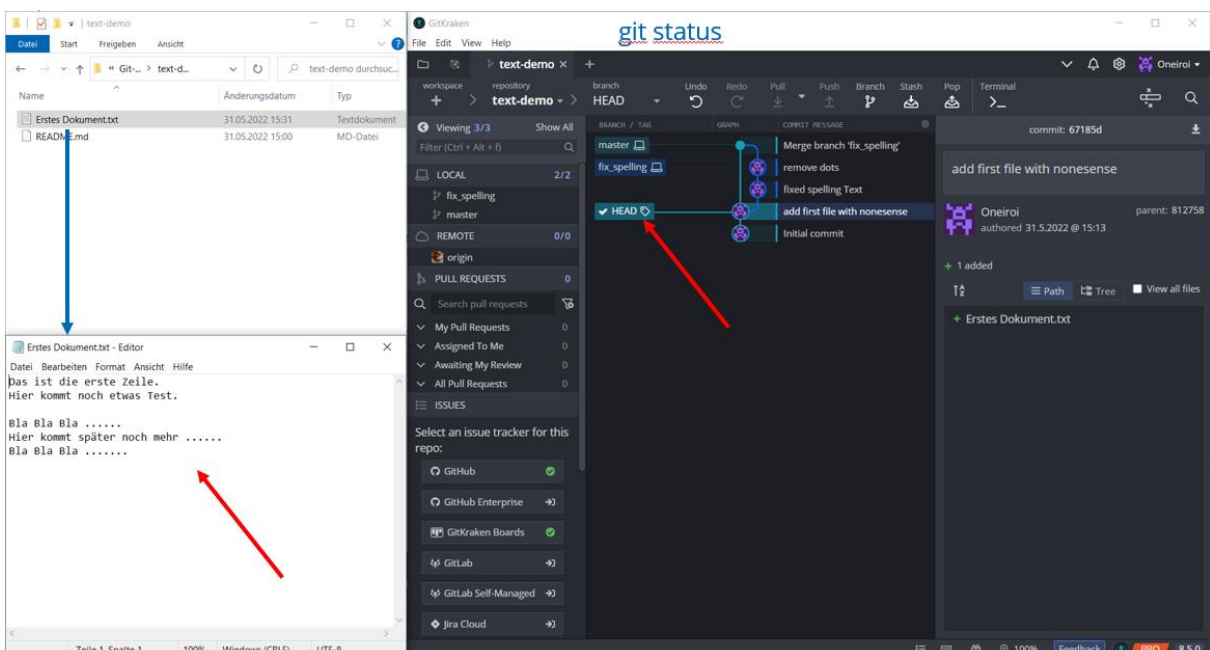
Mit „git checkout master“ wird wieder in die Branch „master“ gewechselt. Dieser befindet sich noch auf dem Stand des letzten darin durchgeführten Commits, in diesem Fall „add first file with nonsense“. Mit „git merge <Zweig>“ kann man eine Branch mit einer anderen zusammenfügen. In diesem Fall wird die Branch „fix\_spelling“ in „master“ gemerged. Dieser Vorgang bekommt ebenfalls eine SHA-1-Prüfsumme „bac47f“. Nun zeigt die Branch „master“ auf den letzten Commit „remove dots“.



Mit „git checkout <Commit-ID>“ kann zu einem spezifischen Commit gewechselt werden. Um aus „add first file with nonsense“ zu wechseln, müsste man „git checkout 67185d“ angeben. Beginnend wird die SHA-1-Prüfsumme kopiert und anschließend „checkout this commit“ gewählt.

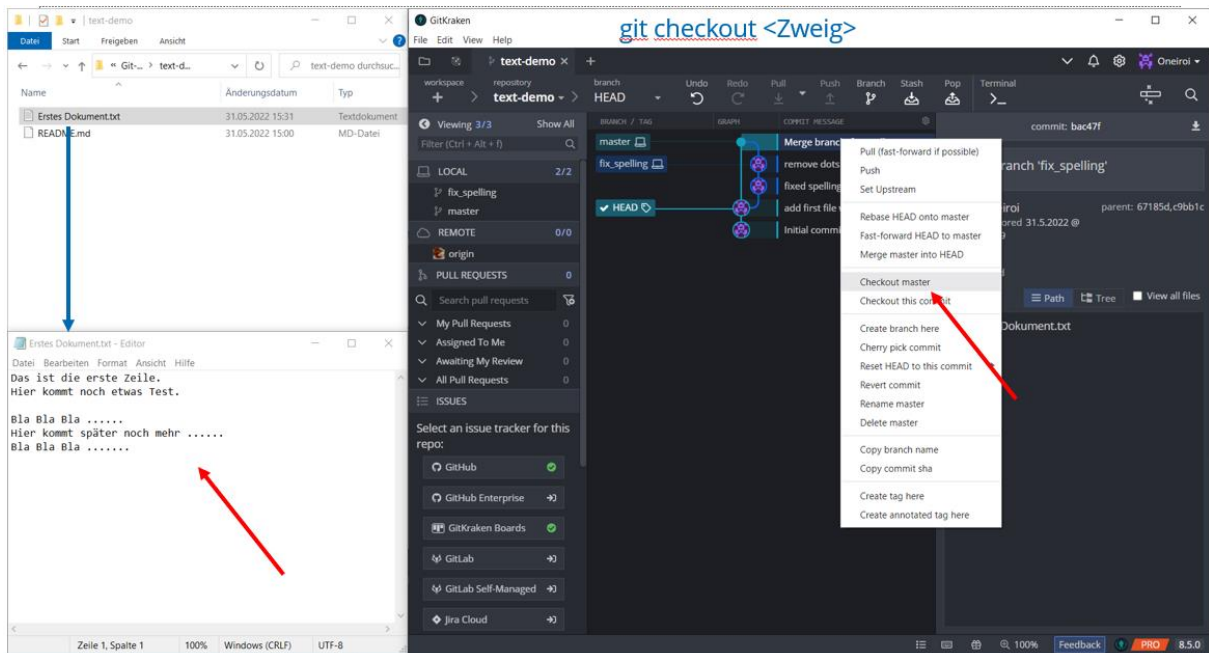


Nun wird der Zeiger „HEAD“ auf den Commit „add first file with nonsense“ gesetzt. Dieser Zeiger zeigt immer auf den lokalen Branch, auf dem man sich gegenwärtig befindet. Da eben auf einen spezifischen Commit gewechselt wurde, wechselte auch der Zeiger „HEAD“ von „master“ zu dem spezifischen Commit. Mit „git status“ lässt sich wieder der Zustand der Datei anzeigen. Es ist zu sehen, dass dies die erste Version der Datei „Erstes Dokument.txt“ ist.

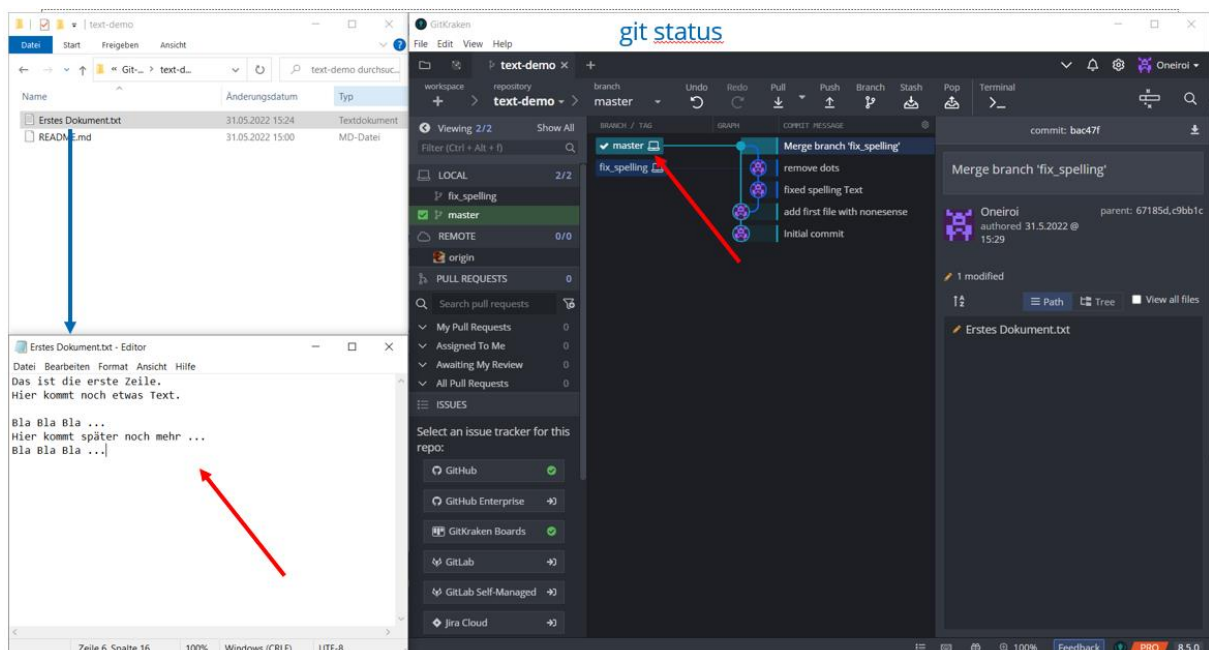




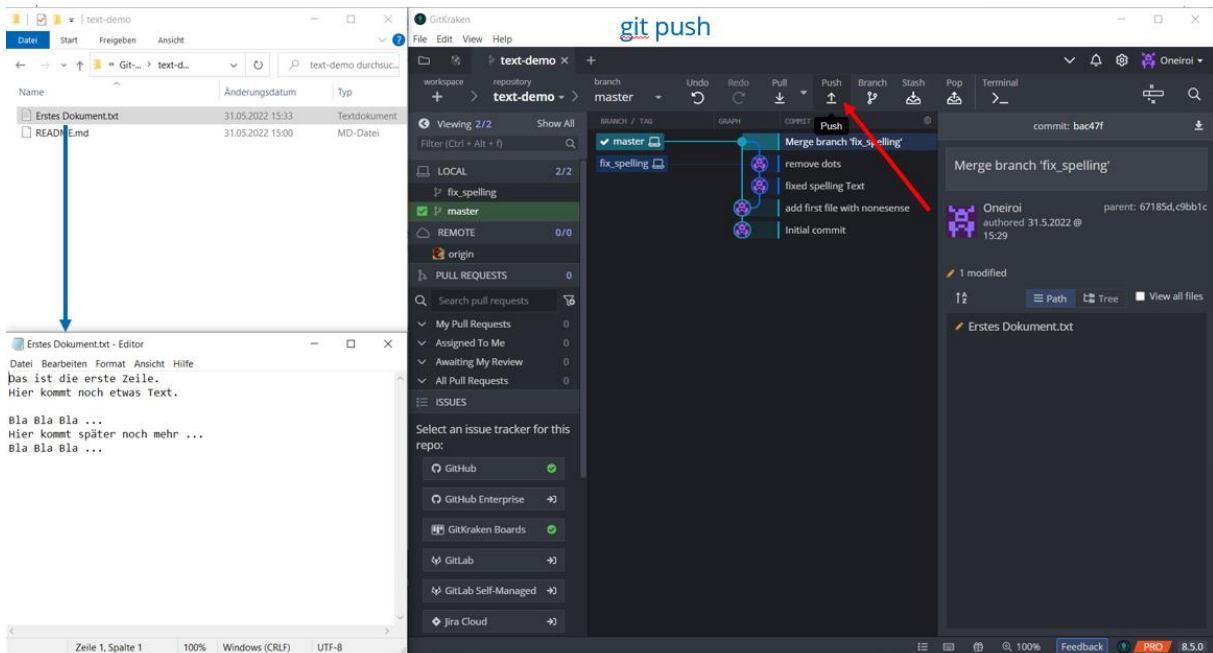
Nun wird wieder in die Branch „master“ gewechselt („git checkout master“). Der Pointer „HEAD“ zeigt nun wieder auf die Branch „master“, da man sich nun darin befindet.



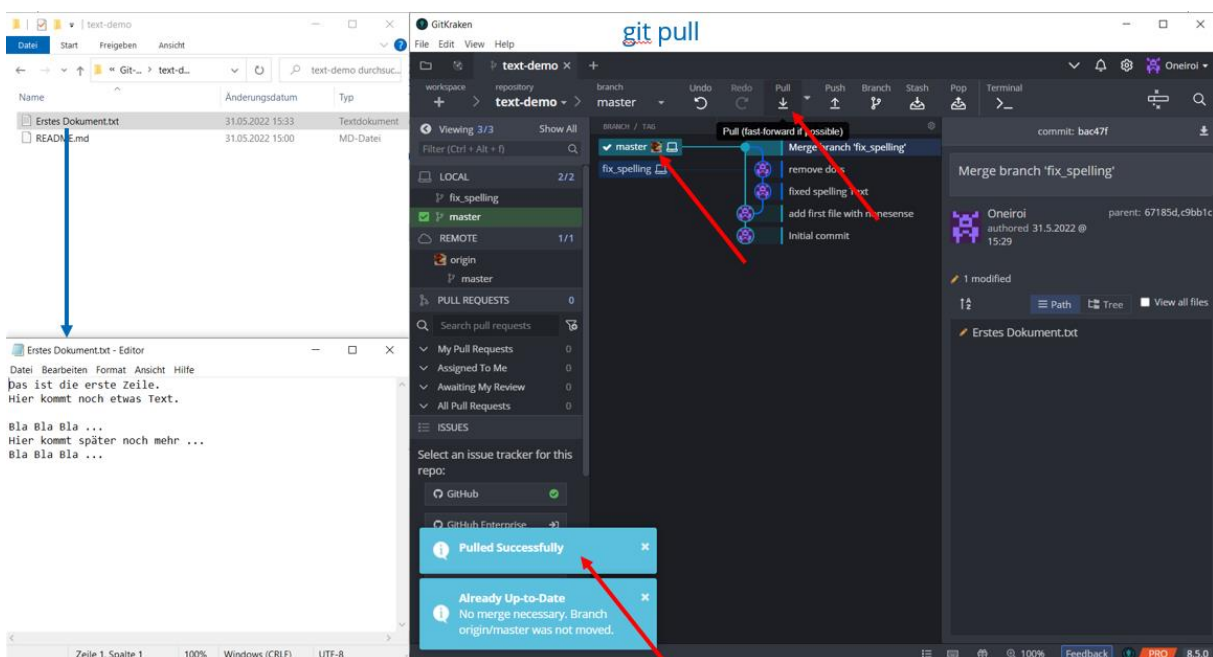
Gibt man sich den Zustand der Dateien aus, ist zu erkennen, dass „Erstes Dokument.txt“ nun den Status modified hat. Links ist zu erkennen, dass es sich wieder um das bearbeitete Dokument handelt, da die Branch „master“ auf den letzten Commit zeigt.



Mit „git push“ können Branches auf einen remote Server hochgeladen werden, sodass sie für andere zugänglich sind. In diesem Fall wird die Branch „master“ gepushed.

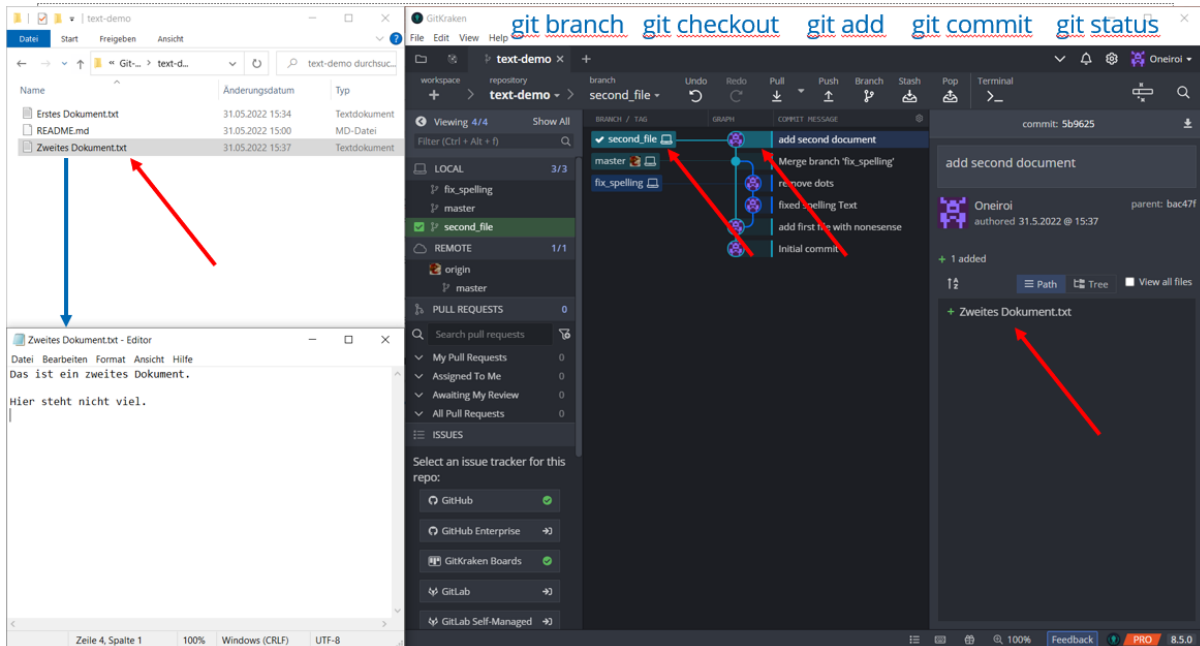


Mit „git pull“ wird auf dem Server nach der versionierten Branch, in diesem Fall „master“, gesucht. Dann wird diese mit der aktuellen Branch verzweigt, die Inhalte vom Server heruntergeladen und anschließend wird versucht, die Remote-Branch zu mergen.



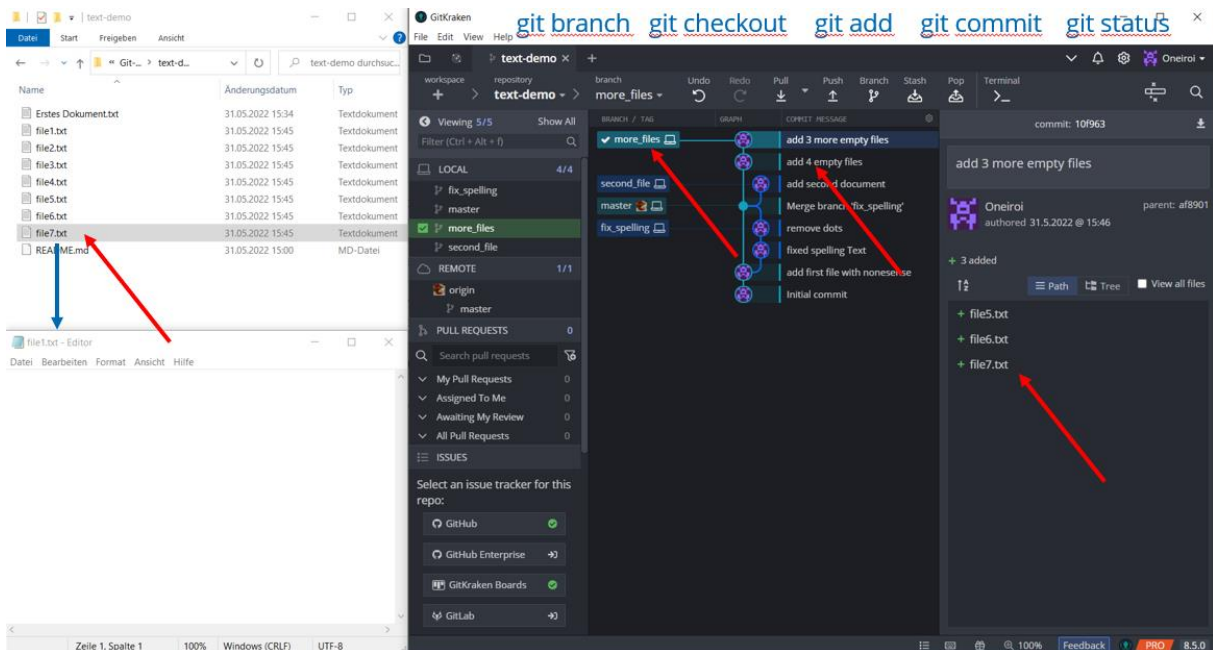


Als nächstes wird eine neue Branch „second\_file“ erstellt und in diese gewechselt. Es wird auf dem Computer eine zweite Datei „Zweites Dokument.txt“ erstellt. Dieses wird auf den Zustand „staged“ gesetzt, sodass es für den nächsten Commit vorgemerkt ist. Die Datei wird mit der Nachricht „add second document“ zur Branch „second\_file“ committed. Der Zustand des Dokuments kann über „git status“ eingesehen werden. Es wird unter „added“ geführt. Die Branch „second\_file“ zeigt jetzt auf den letzten Commit, somit auf „add second document“.

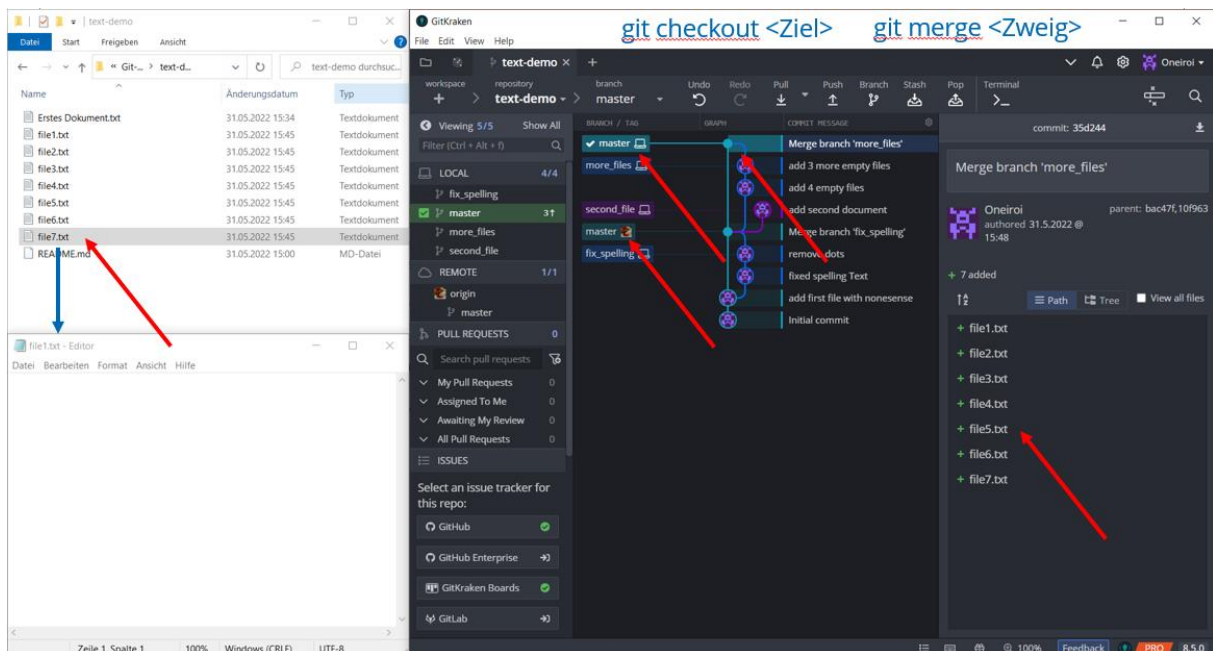


Als nächstes wird eine neue Branch „more\_files“ erstellt und in diese gewechselt. Es werden auf dem Computer die Dateien „file 1.txt“ bis „file 7.txt“ erstellt. „file 1.txt“ bis „file 4.txt“ werden geadded und somit auf den Zustand „staged“ gesetzt, sodass sie für den nächsten Commit vorgemerkt sind. Sie werden mit der Nachricht „add 4 empty files“ in der Branch „more\_files“ comitted. Die Branch „more\_files“ zeigt nun auf diesen Commit.

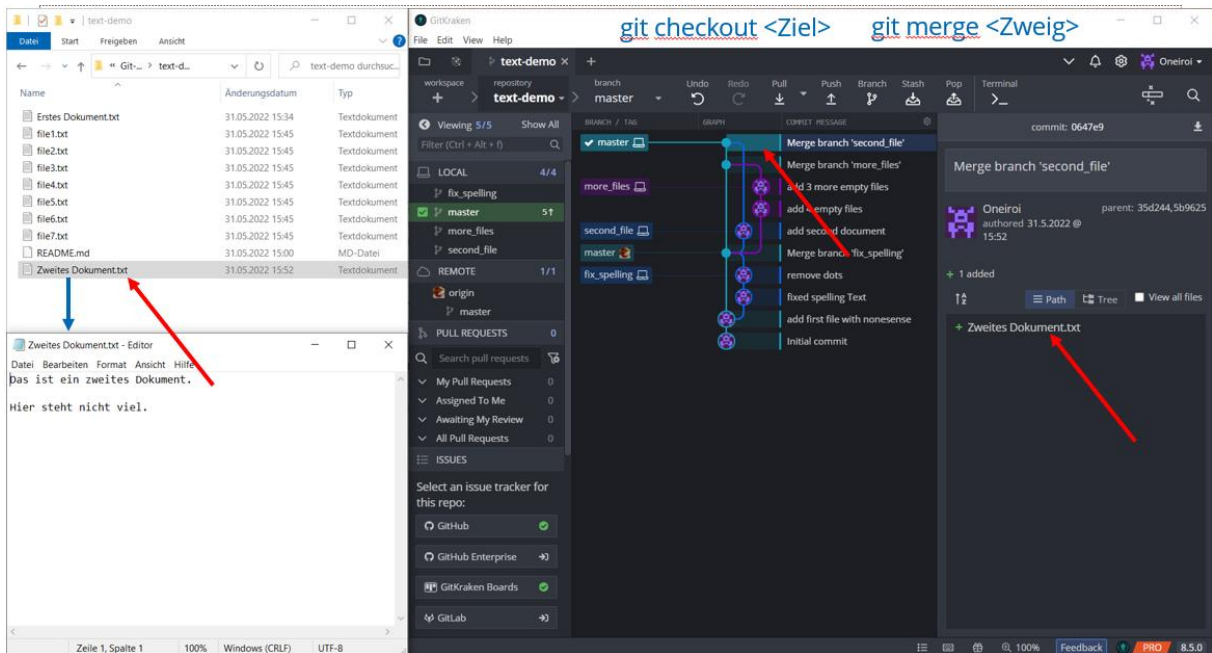
Im Anschluss werden „file 5.txt“ bis „file 7.txt“ geadded und somit auf den Zustand „staged“ gesetzt, sodass sie für den nächsten Commit vorgemerkt sind. Sie werden mit der Nachricht „add 3 more empty files“ in der Branch „more\_files“ comitted. Die Branch „more\_files“ zeigt nun auf diesen Commit. Der Zustand der Dateien wird als „added“ angezeigt.



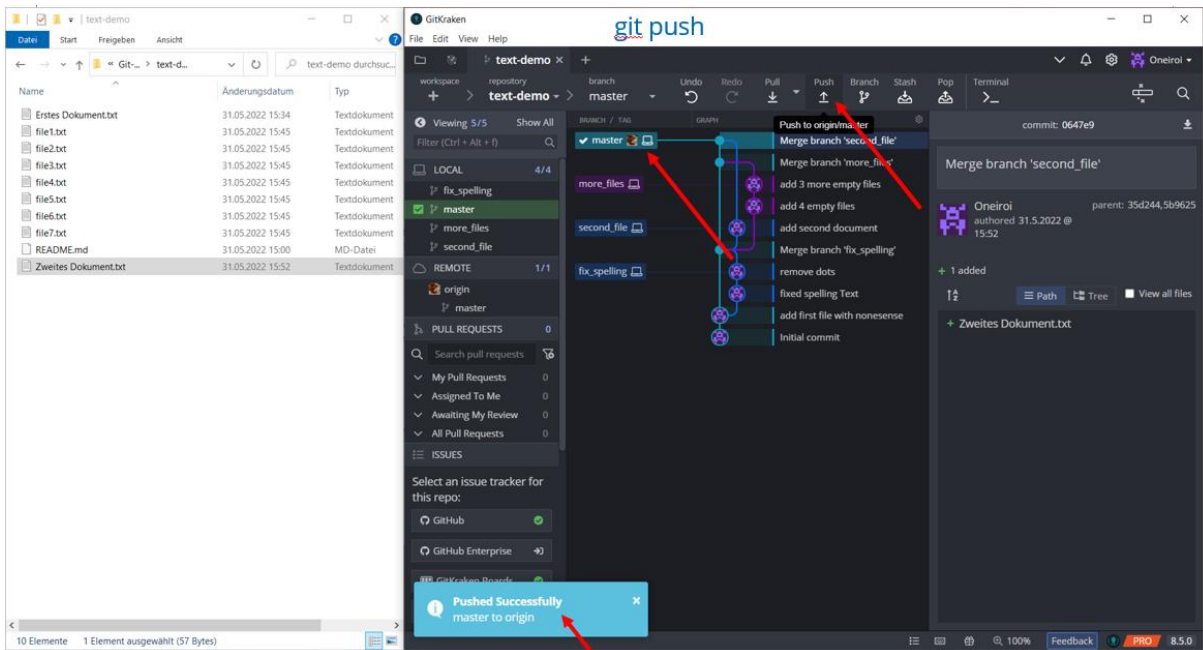
Nun wird in die Branch „master“ gewechselt und die Branch „more\_files“ gemerged. Die Branch „master“ zeigte vorher auf den letzten Commit „remove dots“. Nun zeigt sie auf den neuen letzten Commit „add 3 more empty files“. Die remote verfügbare Branch „master“ ist weiterhin auf dem alten Stand.



Nun wird die Branch „second\_file“ in die Branch „master“ gemerged. Somit wird die Datei „Zweites Dokument.txt“ in die Branch hinzugefügt.

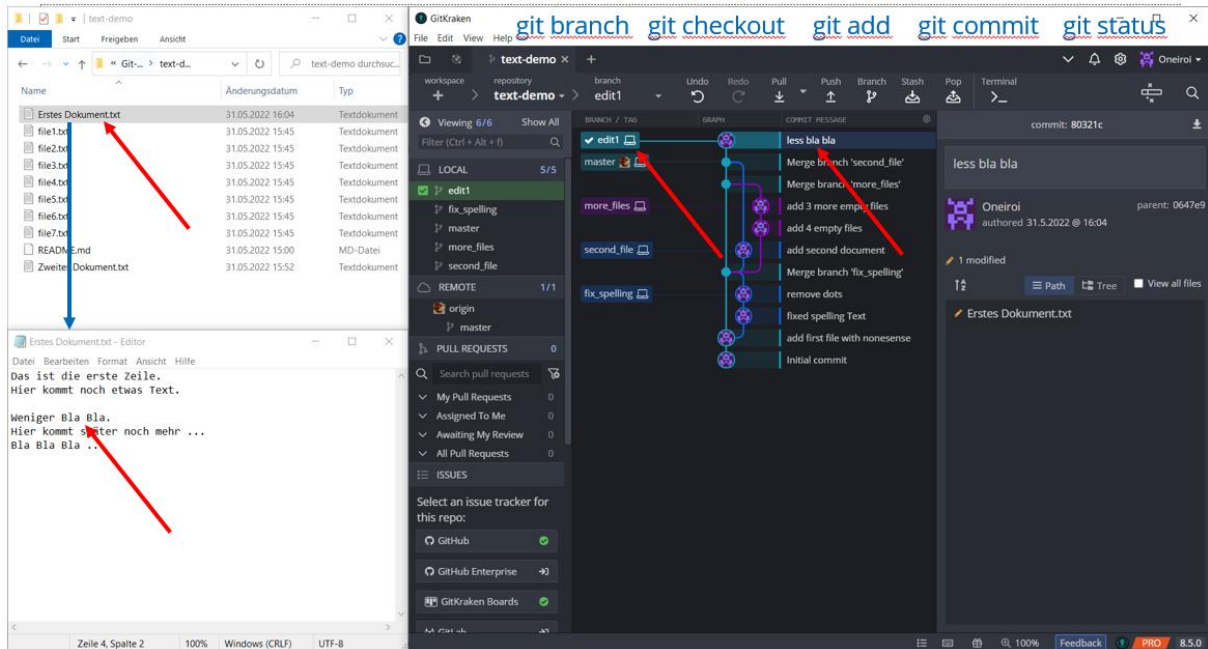


Anschließend wird die Branch „master“ erneut gepushed. Somit sind die remote-Variante der Branch „master“ und die lokale Variante wieder identisch.

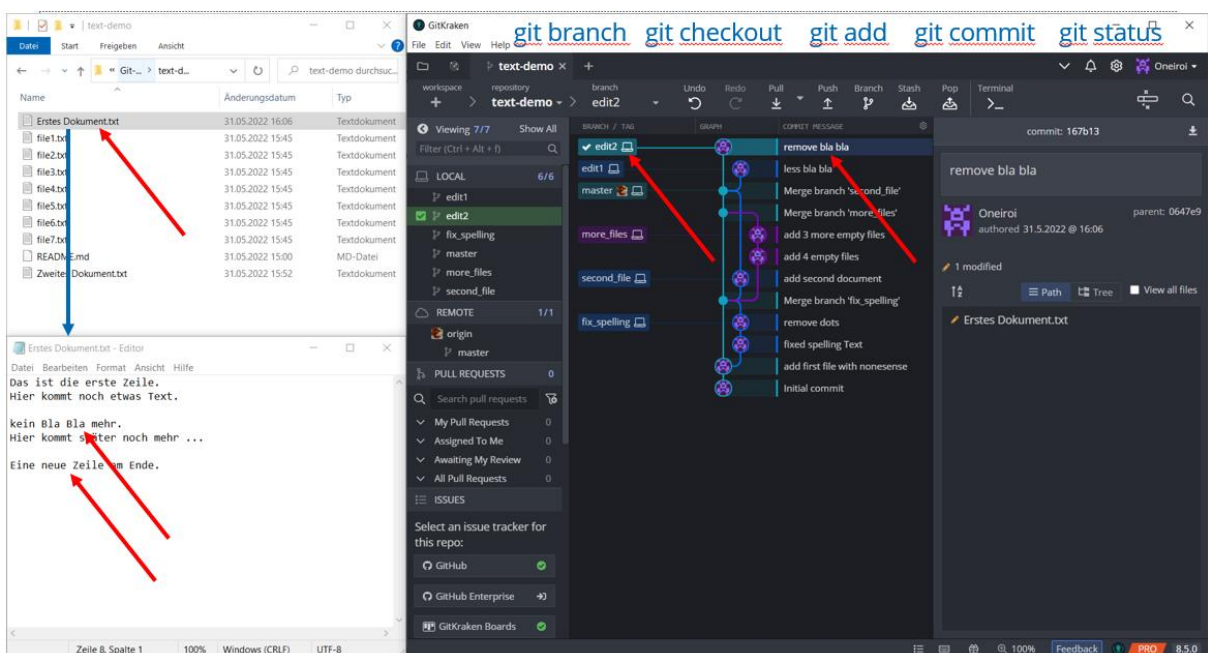


Im Folgenden wird eine neue Branch „edit1“ erstellt und in diese gewechselt. Die Datei „Erstes Dokument.txt“ wird geändert. Diese Version der Datei wird in der Branch „edit1“ hinzugefügt und die Datei

somit auf den Zustand „staged“ gesetzt, sodass sie für den nächsten Commit vorgemerkt ist. Mit dem Kommentar „less bla bla“ wird die Datei in der Branch „edit1“ committed. Der Status dieser Datei wird hier auf modified gesetzt. Die Branch zeigt nun auf den Commit „less bla bla“.

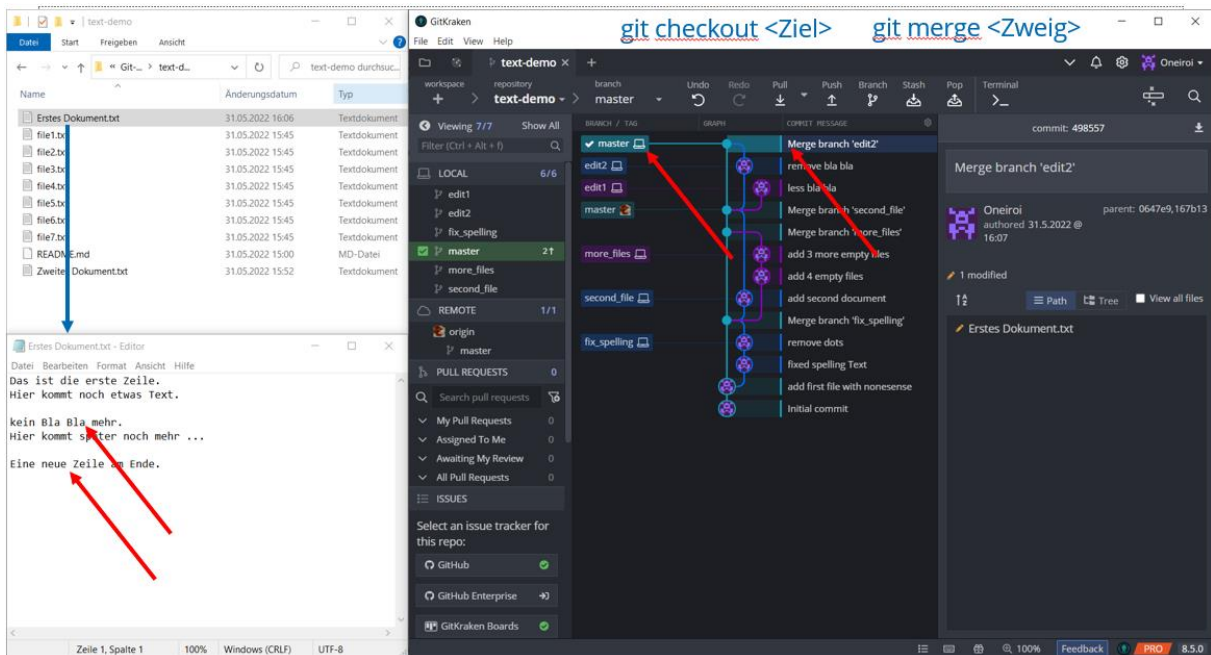


Anschließend wird eine neue Branch „edit2“ erstellt und in diese gewechselt. Die Datei „Erstes Dokument.txt“ wird geändert. Diese Version der Datei wird in der Branch „edit2“ hinzugefügt und die Datei somit auf den Zustand „staged“ gesetzt, sodass sie für den nächsten Commit vorgemerkt ist. Mit dem Kommentar „remove bla bla“ wird die Datei in der Branch „edit2“ committed. Der Status dieser Datei wird hier auf modified gesetzt. Die Branch zeigt nun auf den Commit „remove bla bla“.

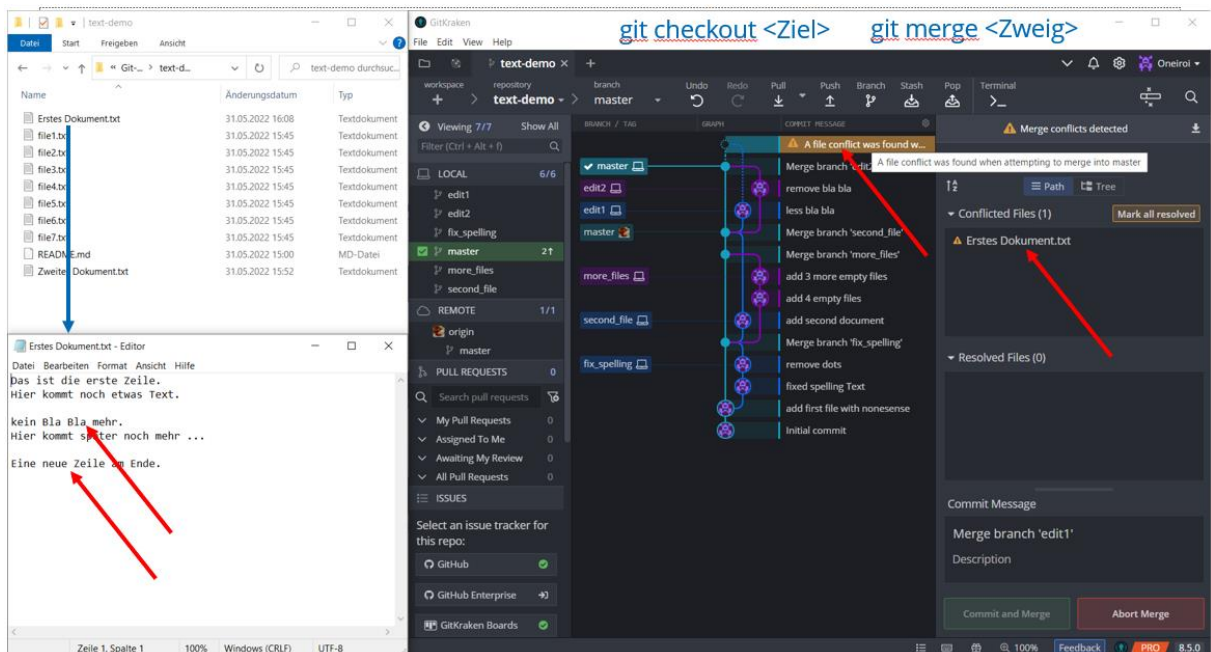




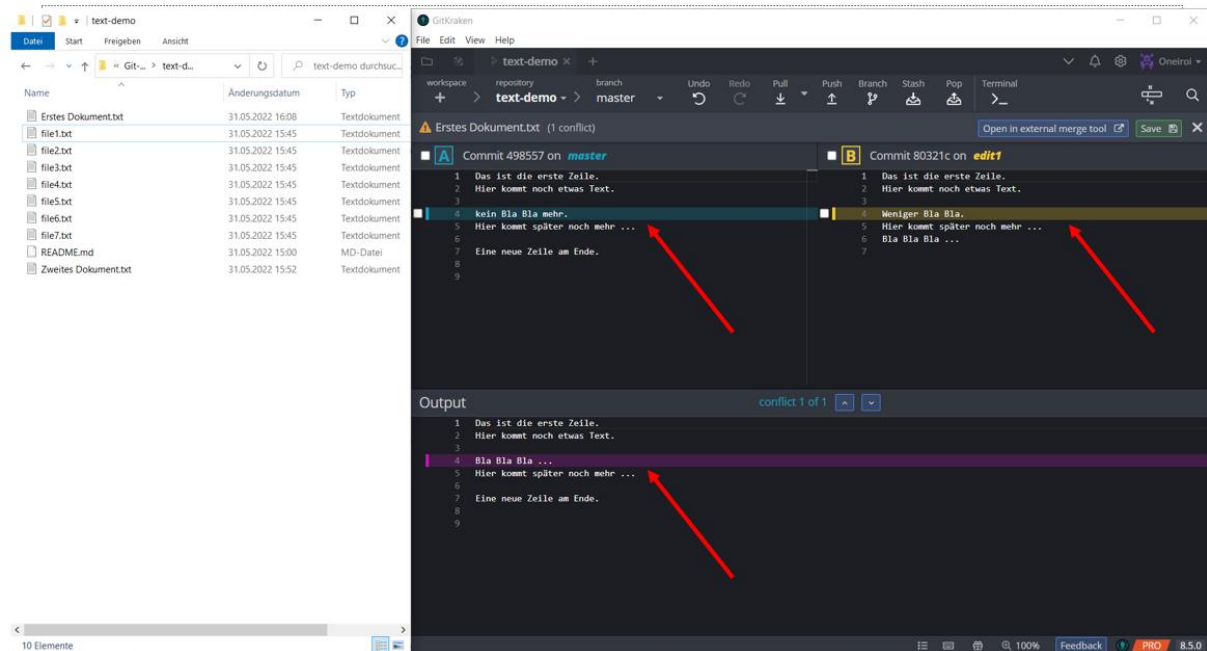
Nun wird wieder in die Branch „master“ gewechselt und die Branch „edit2“ gemerged.



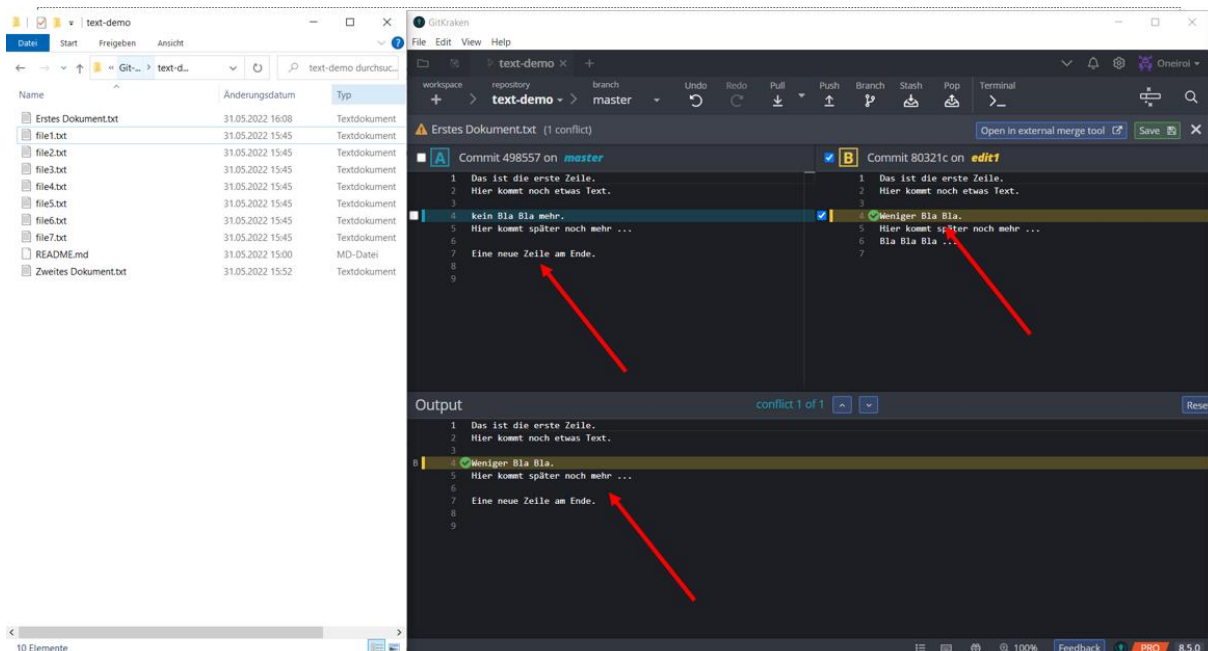
Nun wird versucht, auch die Branch „edit1“ zu mergen, wobei es zu einem file conflict im Dokument „Erstes Dokument.txt“ kommt.



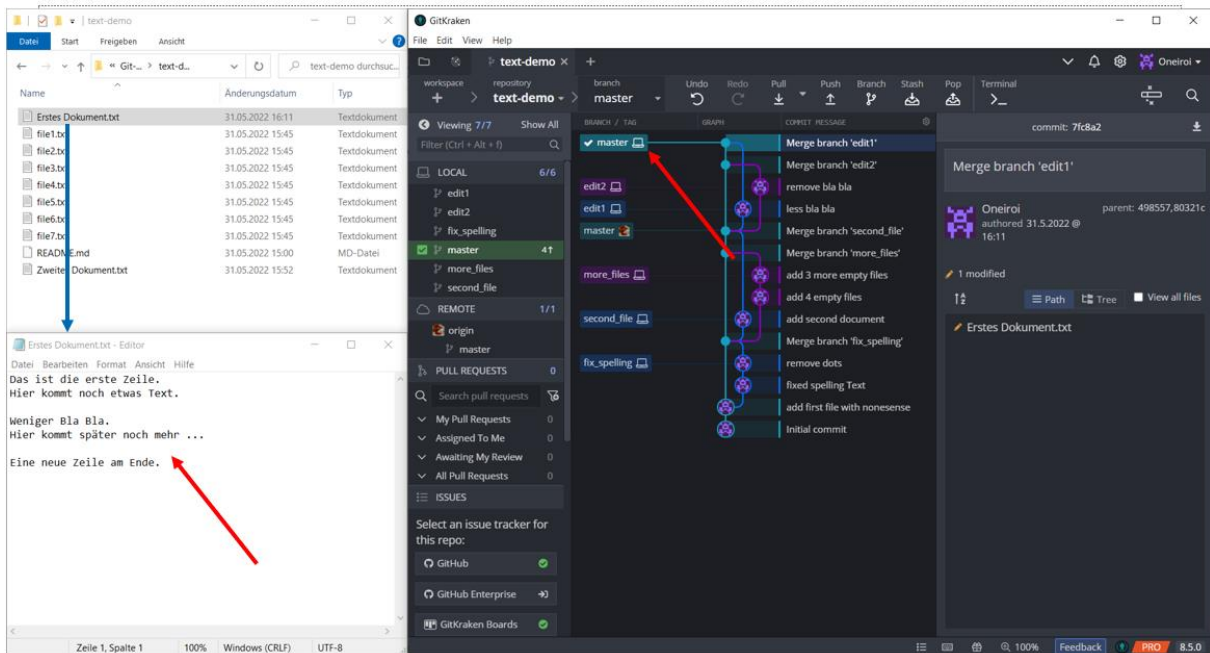
Der Konflikt entstand in Zeile 4 der Datei „Erstes Dokument.txt“. In der Branch master steht in dieser Zeile der Text „kein Bla Bla mehr.“, da der Master auf den letzten Commit zeigt, welcher „remove bla bla“ ist, wohingegen in der Branch edit1 dort „Weniger Bla Bla.“ steht. Als Konfliktlösung wird als Output für diese Zeile „Bla Bla Bla ...“ vorgeschlagen.



Möchte man jedoch den Text der Datei „Erstes Dokument.txt“ aus Branch „edit1“ haben, wählt man diese aus.

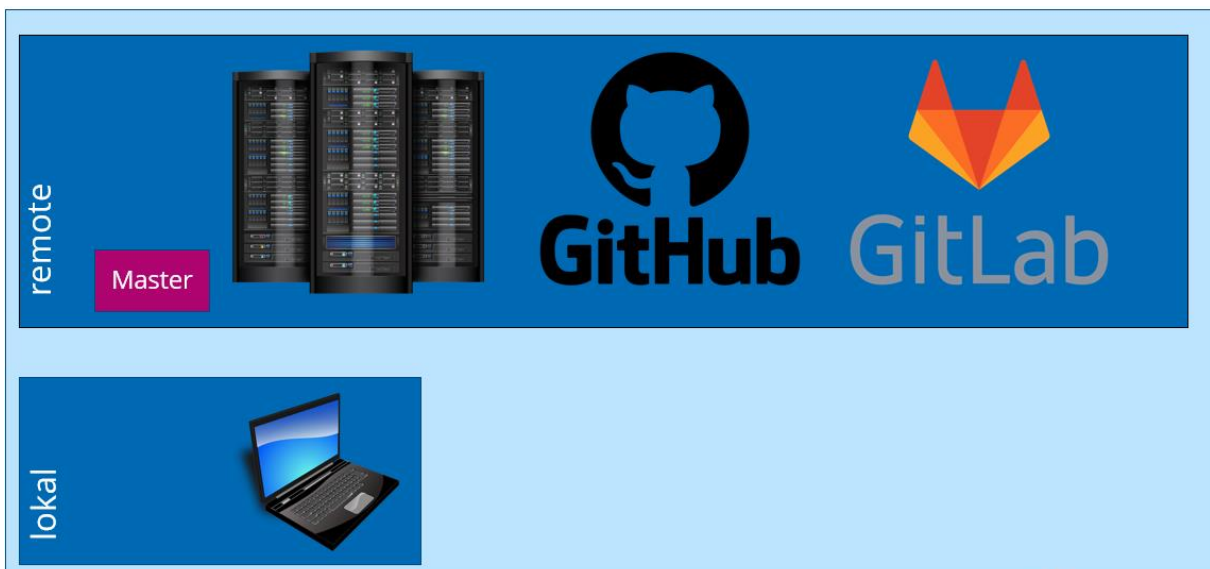


Somit kann die Branch „edit1“ erfolgreich in die Branch „master“ gemerged werden. Der Inhalt der Datei „Erstes Dokument.txt“ ist nun links zu sehen.

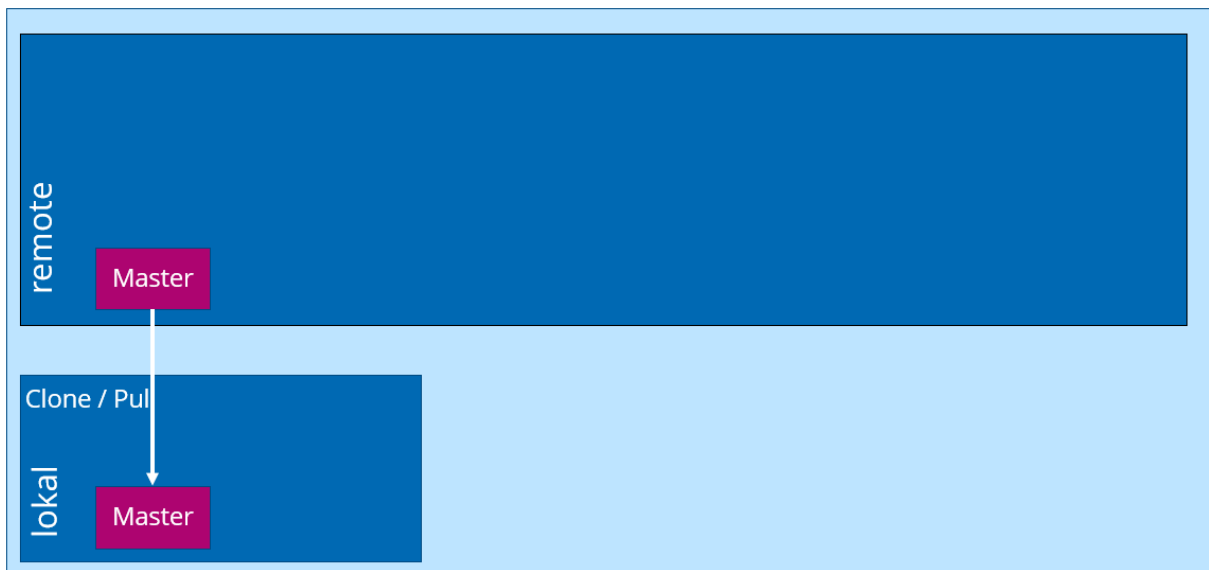


### 1.8.1.2 Git kooperatives Arbeiten intern

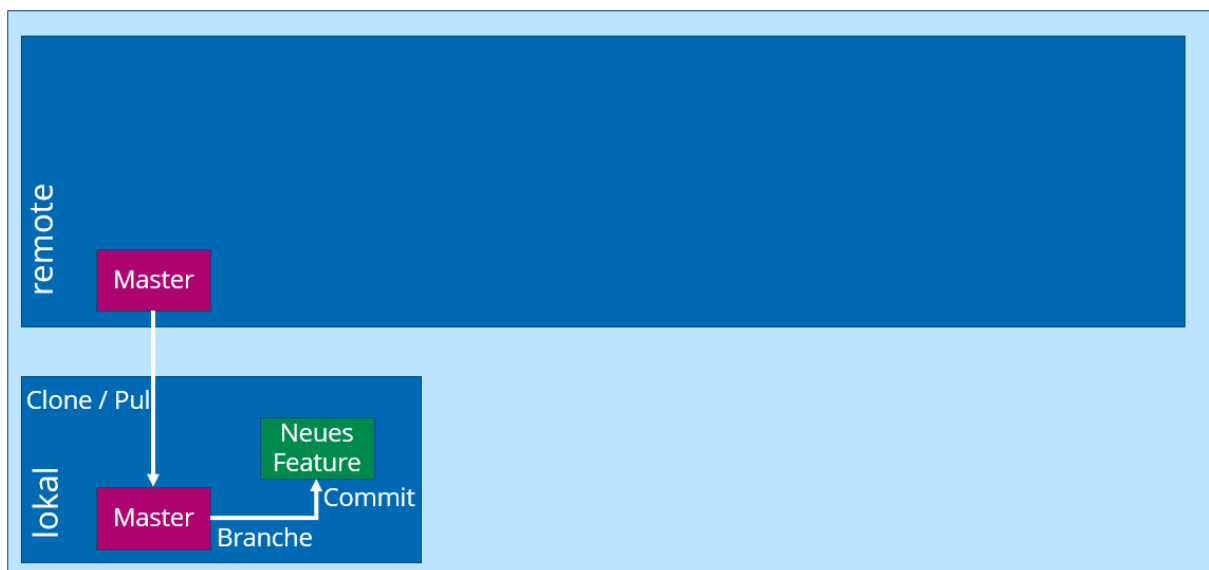
Git steht sowohl remote über einen Server, GitHub oder GitLab als auch lokal auf dem eigenen PC zur Verfügung.



Die Master Branch steht remote zur Verfügung. Um lokal auf sie zugreifen zu können, muss diese Branch beginnend geclost werden. Ist sie einmal lokal verfügbar, kann sie mit pull aktualisiert werden.

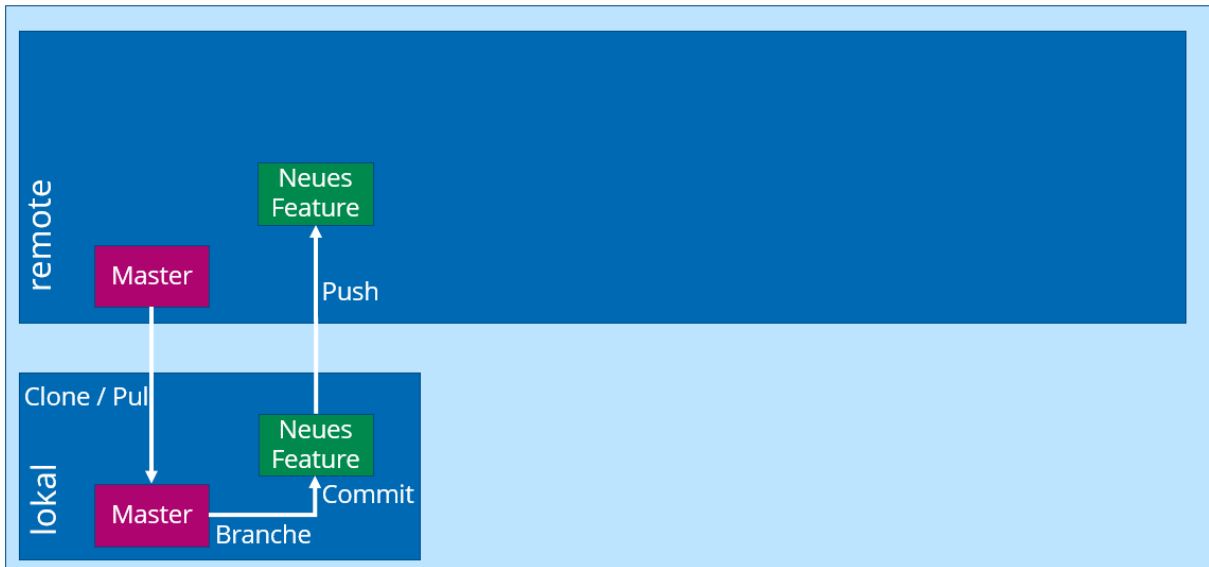


Vom lokalen Master aus kann man Feature-Branche erstellen. In diesen können sich neue Features befinden, welche in die Branch committed werden.

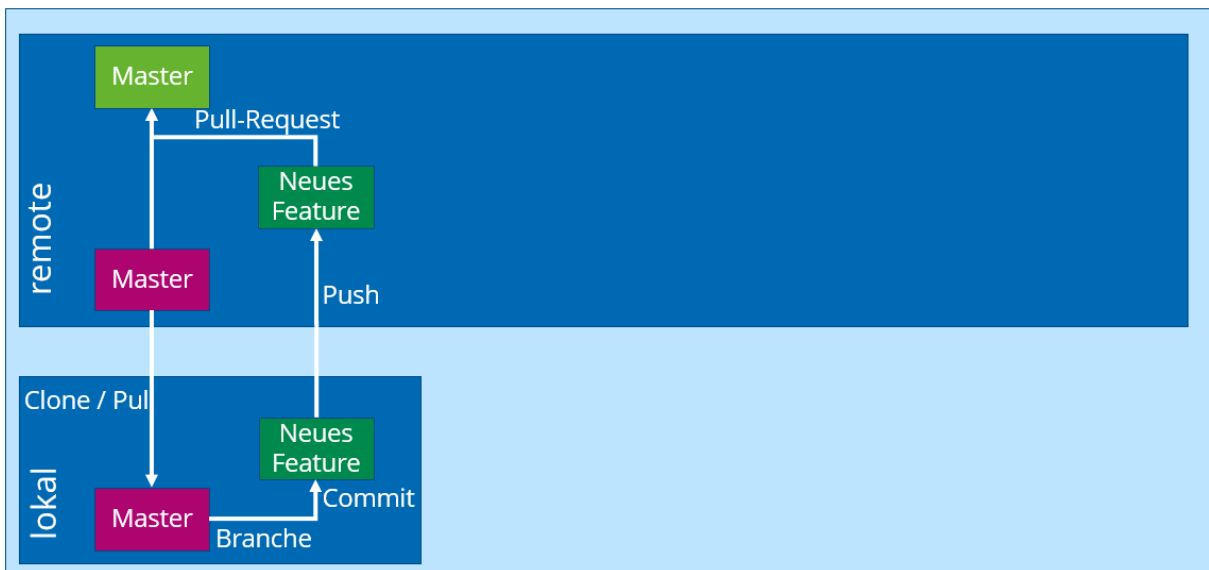


Um diese Änderungen des lokalen Repositorys remote verfügbar zu machen, werden sie gepushed, sodass sie anschließend auch im remote Repository vorliegen.

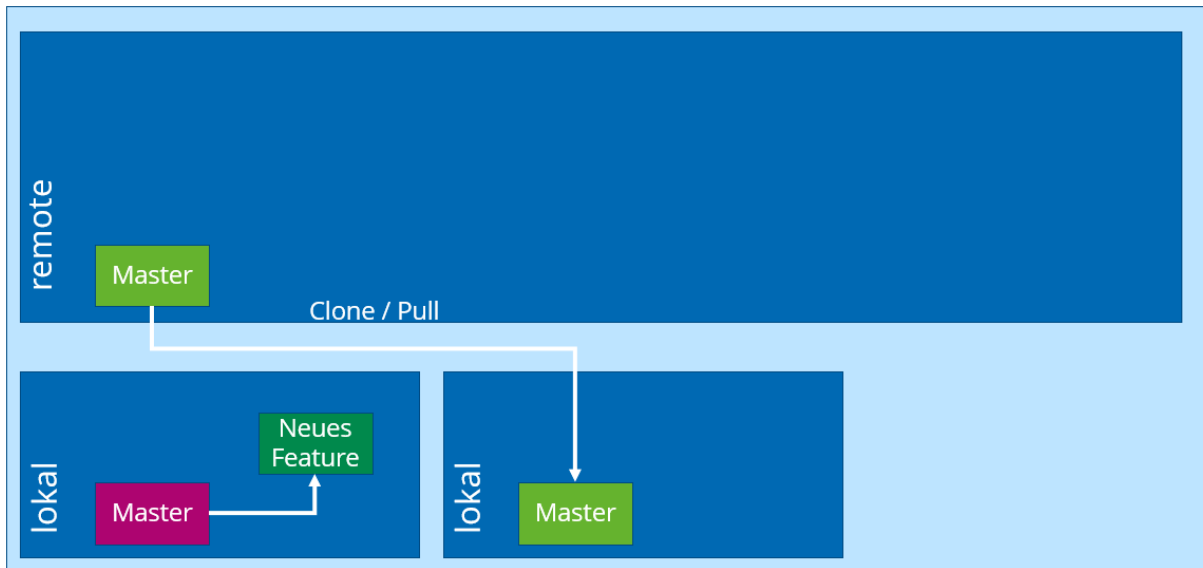




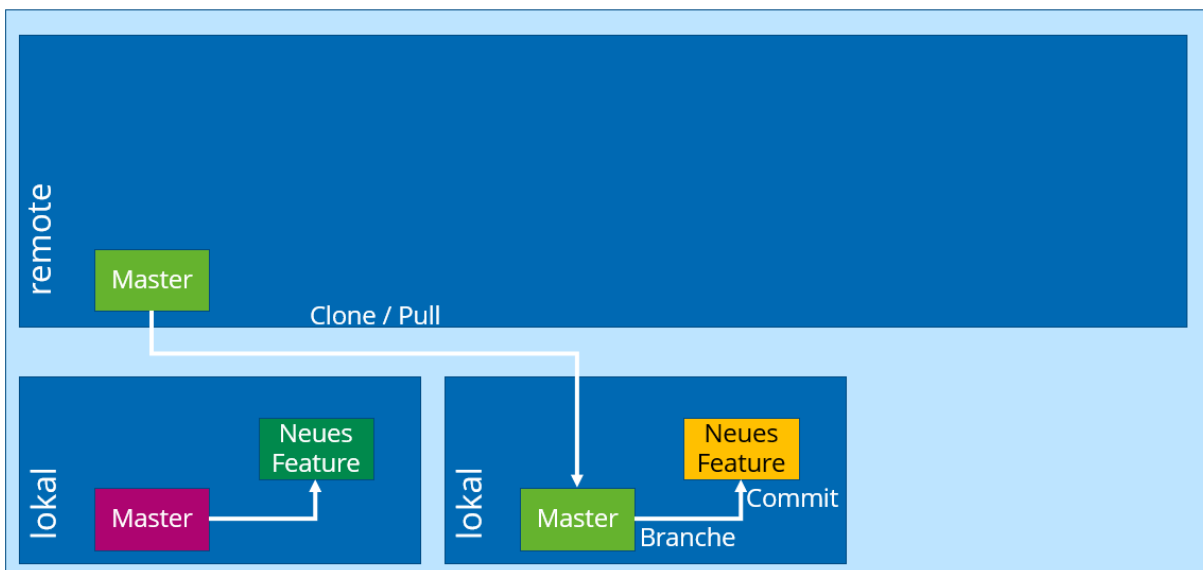
Für das gepushte neue Feature kann eine Pull-request gestellt werden. Hierüber kann den anderen Mitgliedern mitgeteilt werden, dass das Feature fertiggestellt ist. Alle Beteiligten werden darüber informiert, müssen nun den Code prüfen und in den main-Branch (master) mergen.



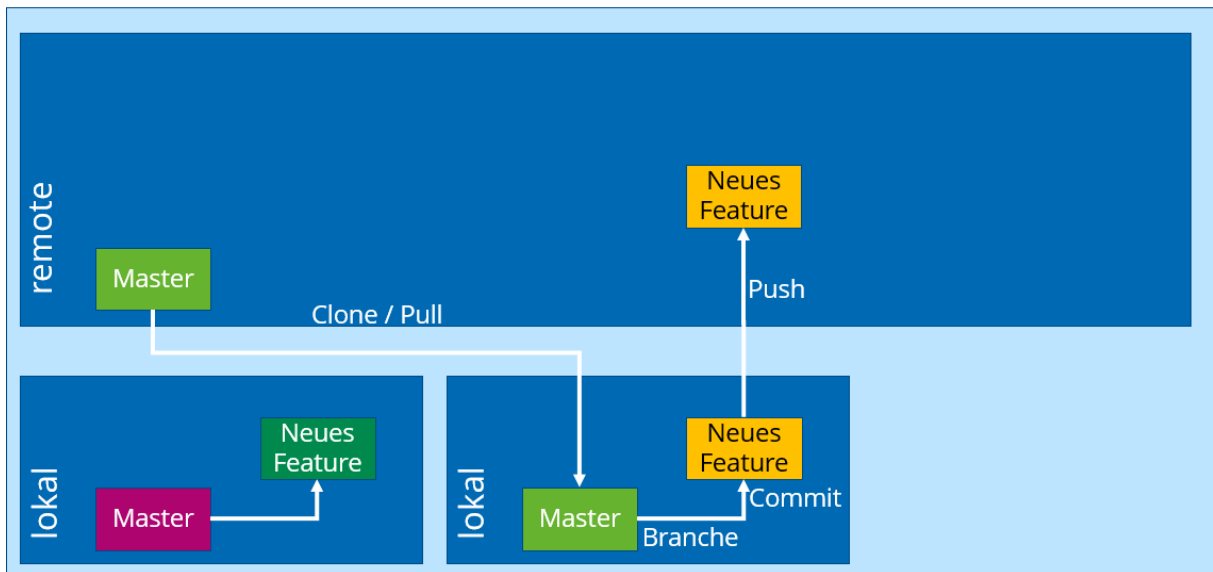
Der aktualisierte Master kann erneut gecloned werden, sodass er lokal für einen anderen Nutzer zur Verfügung steht. Anschließende Aktualisierungen würden mit pull erfolgen.



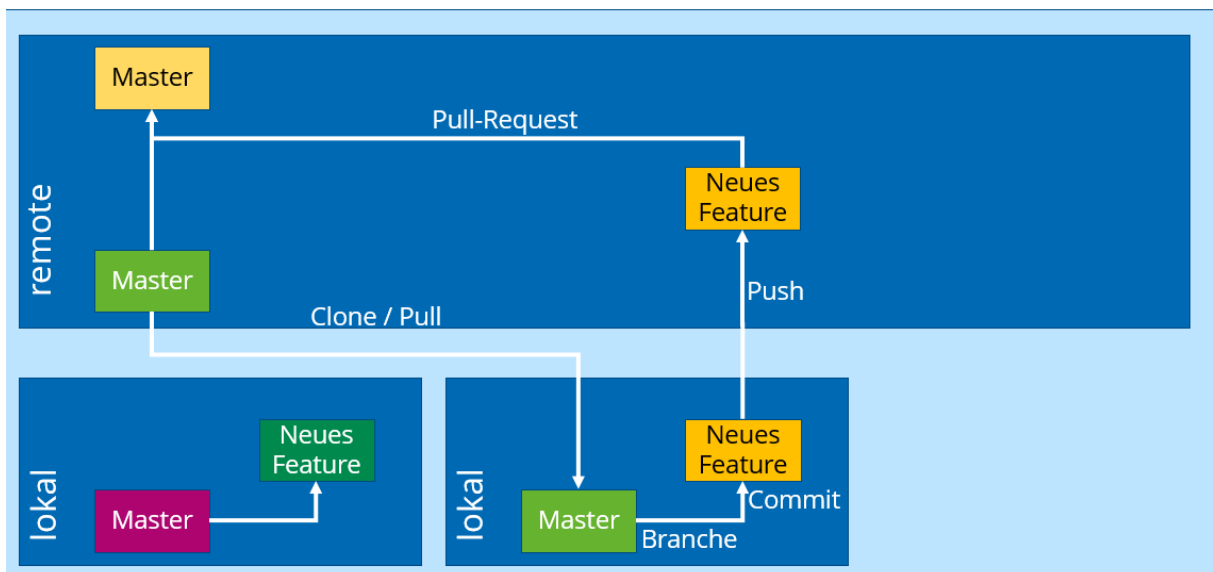
Das weitere Vorgehen erfolgt wie bereits beschrieben. Der Nutzer erstellt lokal eine neue Feature-Branch.



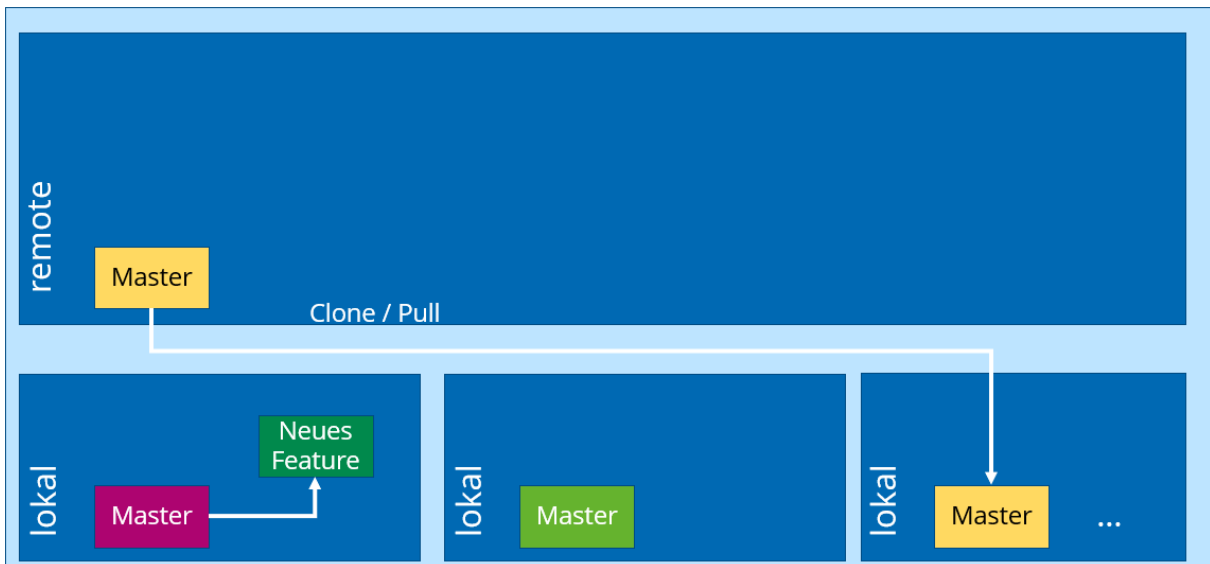
Um diese Änderungen des lokalen Repositorys remote verfügbar zu machen, werden sie gepushed, sodass sie anschließend auch im remote Repository vorliegen.



Für das gepushte neue Feature kann eine Pull-request gestellt werden. Hierüber kann den anderen Mitgliedern mitgeteilt werden, dass das Feature fertiggestellt ist. Alle Beteiligten werden darüber informiert, müssen nun den Code prüfen und in den main-Branch (master) mergen.

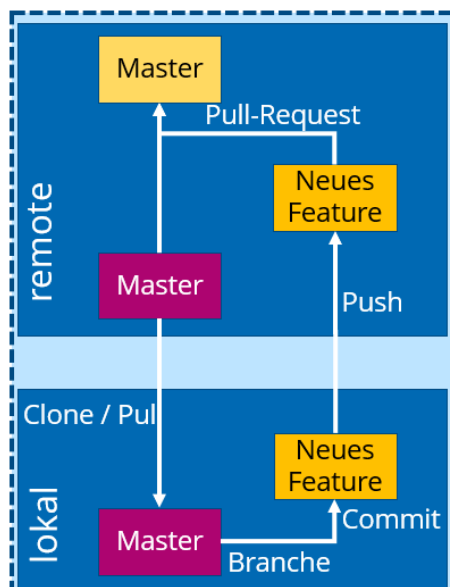


Der aktualisierte Master kann erneut gecloned werden, sodass er lokal für einen anderen Nutzer zur Verfügung steht. Anschließende Aktualisierungen würden mit pull erfolgen.

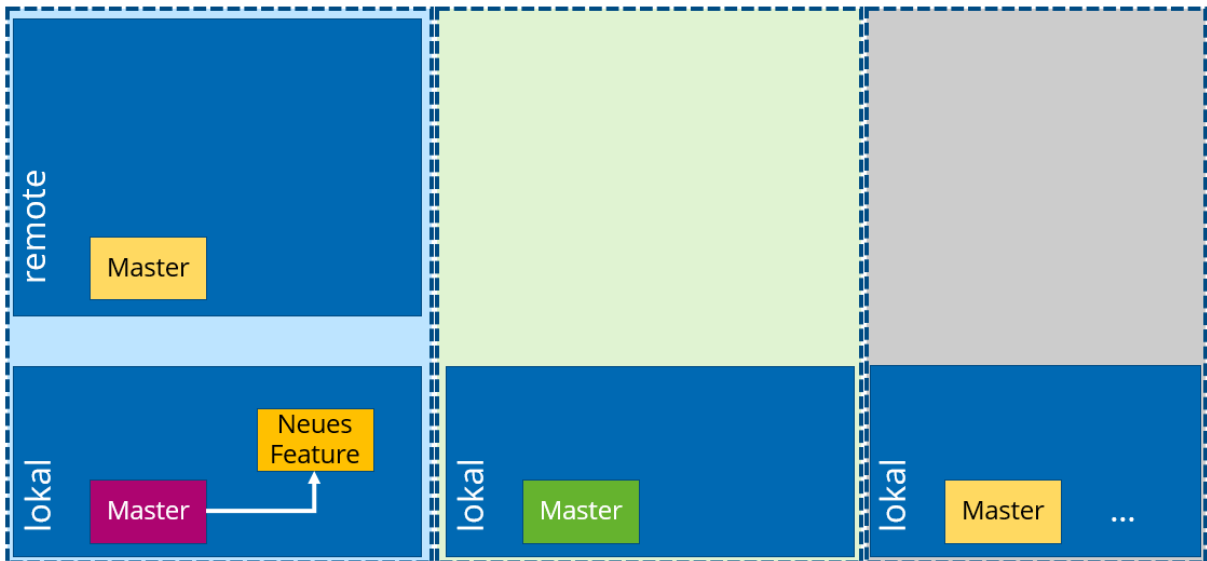


### 1.8.1.3 Git kooperatives Arbeiten extern

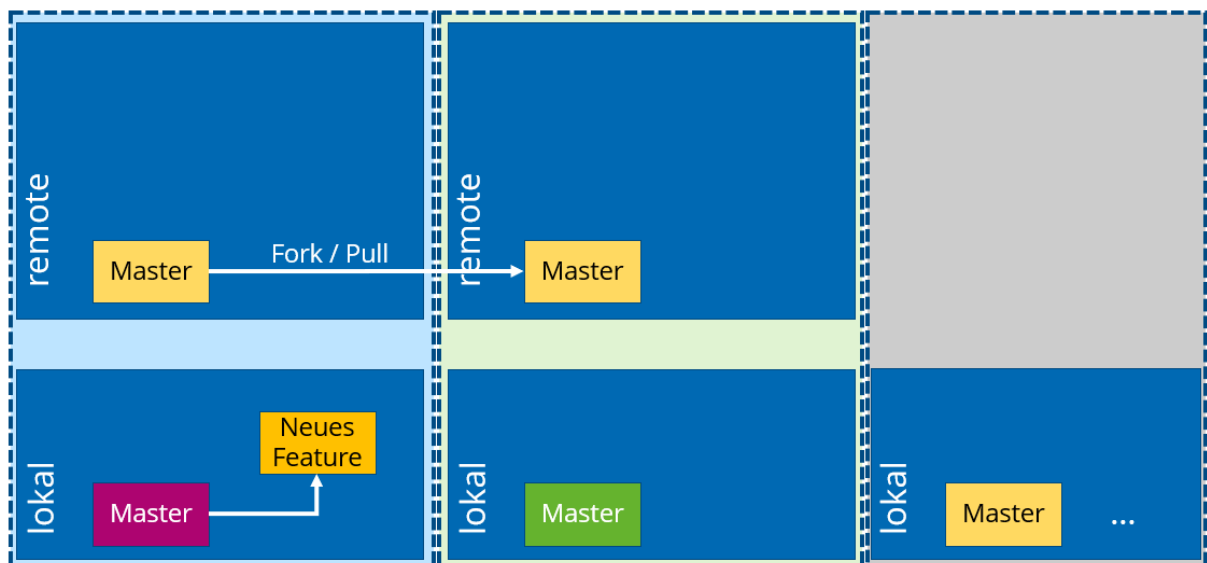
Man kann auch extern mit Git arbeiten. Dies wird mit getrennten Namespaces realisiert. Wie bereits beschrieben kann der remote vorliegende Master mit clone bzw. pull lokal hinzugefügt bzw. aktualisiert werden. Eine lokal angelegte Feature-Branch kann gepushed, damit sie remote zur Verfügung steht. Daraufhin wird eine Pull-Request gestellt. Hierüber kann den anderen Mitgliedern mitgeteilt werden, dass das Feature fertiggestellt ist. Alle Beteiligten werden darüber informiert, müssen nun den Code prüfen und in den main-Branch (master) mergen.



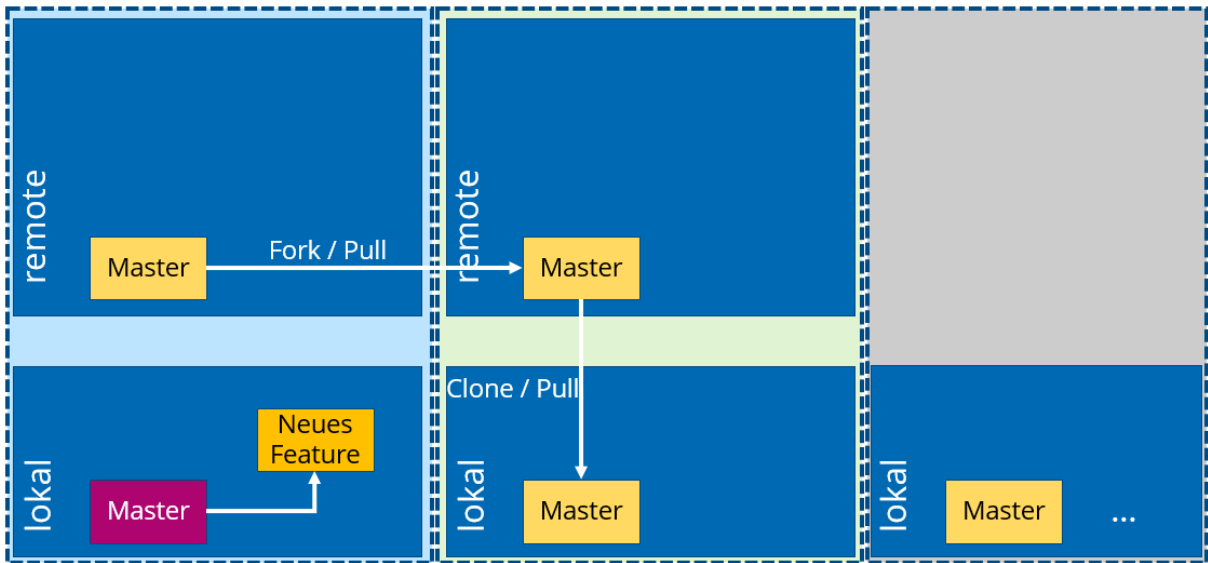
Verschiedene lokale Geräte können verschiedene Versionen des Masters haben. Hierbei handelt es sich um verschiedene Namespaces.



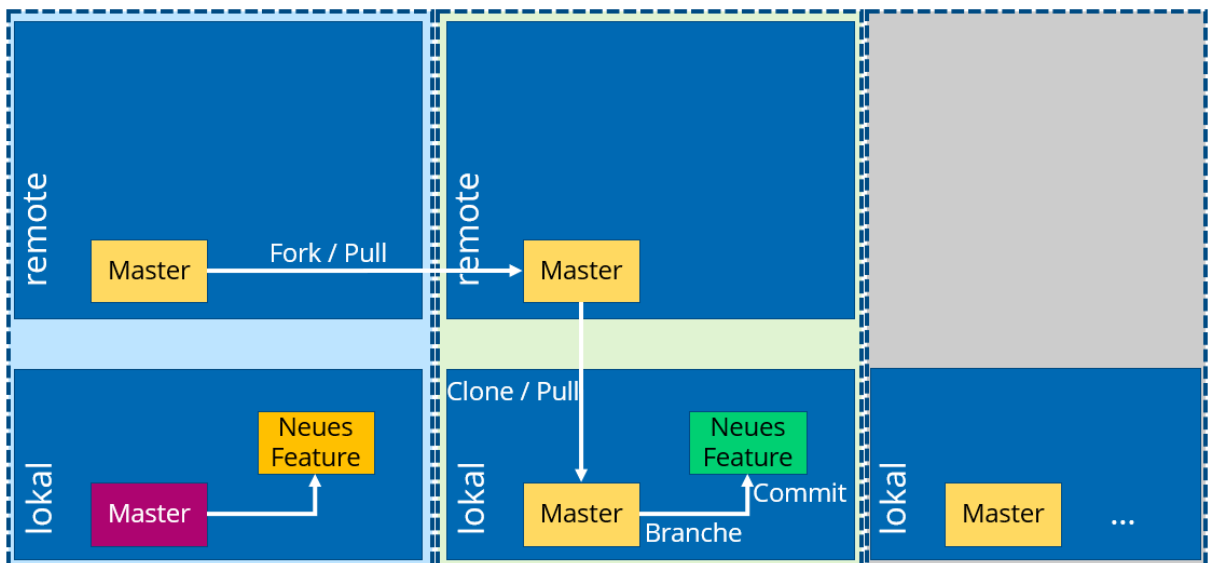
Möchte man in einem Projekt mitwirken, für welches man keine Push-Berechtigungen hat, kann man dieses Projekt „forken“. Somit wird eine Kopie des Projekts erstellt, welche nur für den einen Nutzer ist. Es befindet sich im Namespace des Nutzers und Daten können dorthin hochgeladen werden.



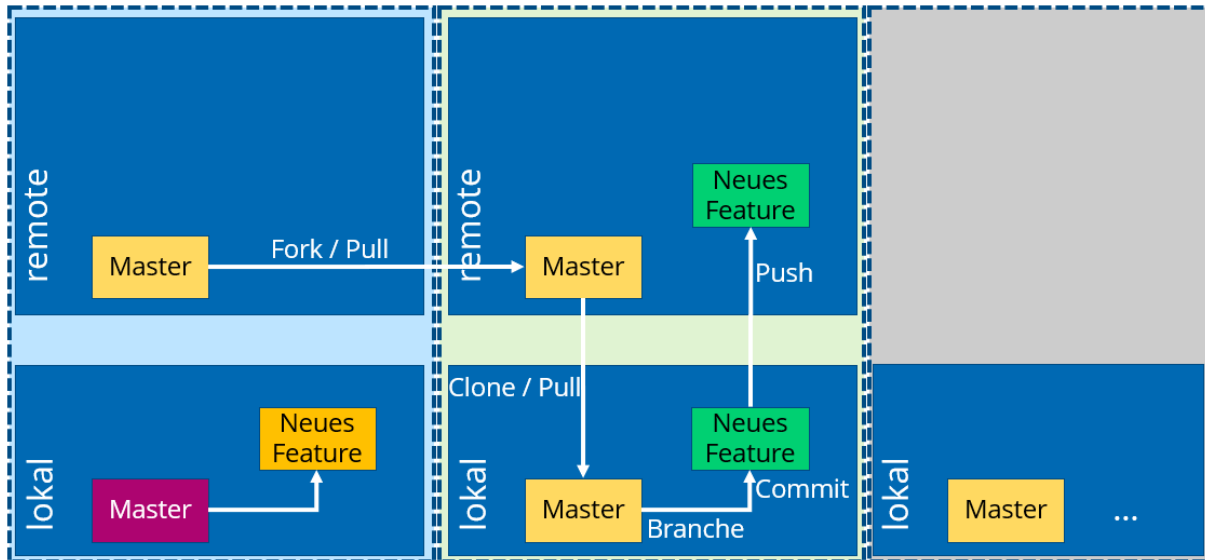
Dieser Kopie der Branch Master kann anschließend wieder mit clone bzw. pull lokal abgespeichert werden, sodass das Projekt auch lokal zur Verfügung steht.



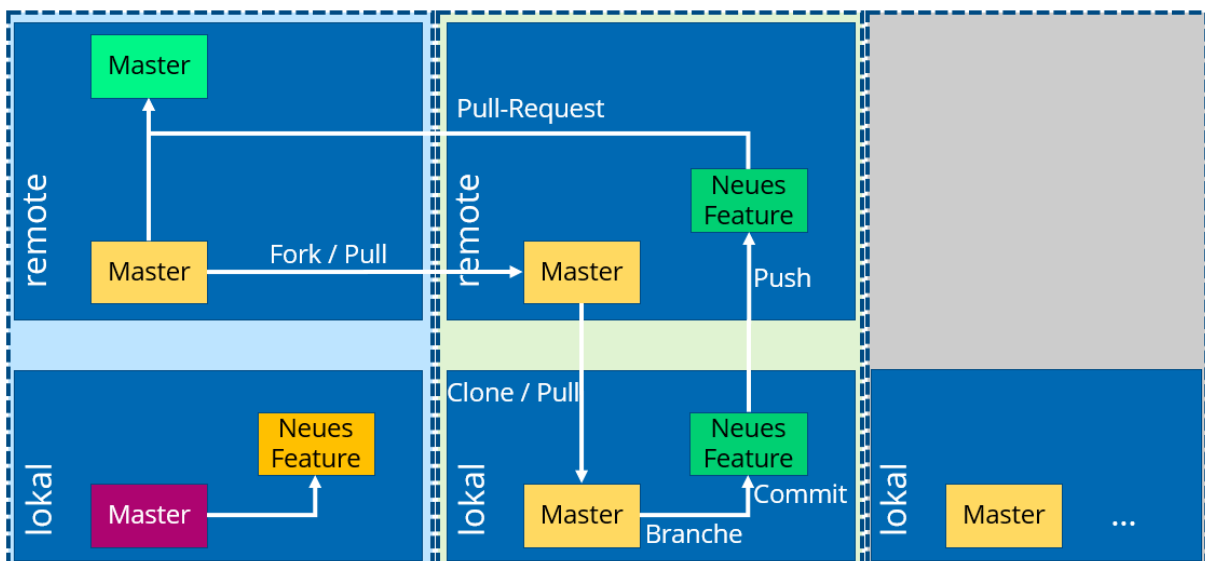
Lokal können neue Feature-Branches erstellt werden.



Anschließend wird diese Branch gepushed, sodass sie im Namespace des Nutzers remote verfügbar ist.



Nun wird eine pull-request an den Hauptnutzer gestellt, damit die Änderungen im originalen Repository eingebracht werden.



#### 1.8.1.4 Git Benutzungsrichtlinien

Jedes Projekt sollte ein eigenes Repository haben. Pro Feature sollte ein Branche erstellt werden. Commits sollten klein gehalten werden und jedem Commit sollte lauffähig sein. Weiterhin sollten Beschreibungen kurz und treffend formuliert werden. Auf den Master sollte kein Push durchgeführt, stattdessen nutzt man einen Pull-Request. Außerdem sollte stets mit einem Pull der aktuelle Stand aktualisiert werden, bevor mit einem Push die Daten hochgeladen werden. Bei Merge-Konflikten sollte sehr

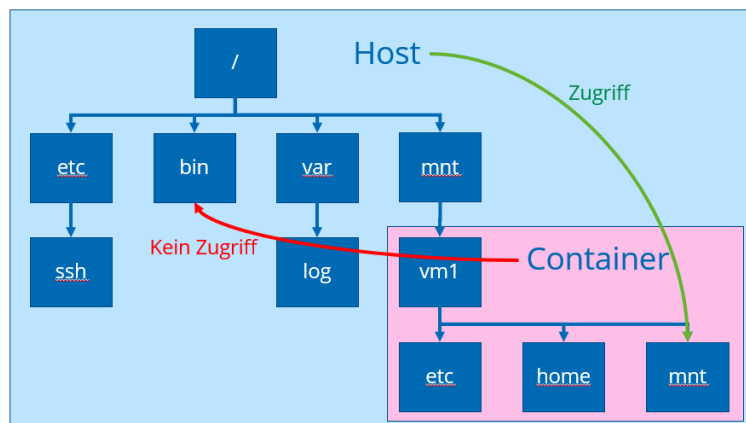


genau gearbeitet und beim ersten Merge-Konflikt eine komplette Kopie erstellt werden, da zu Beginn häufig noch Fehler in der Bedienung gemacht werden. Bei internen Projekten erfolgt ein direkter Zugriff und eine Pull Request. Bei öffentlichen Projekten erfolgt ein indirekter Zugriff durch ein eigenes Remote-Repository welches über Fork erstellt wird.

## 1.8.2 Namespaces

### 1.8.2.1 Abgrenzung

Namespaces realisieren Abgrenzungen. Die einfachste Abgrenzung kann mit `unshare -m` vorgenommen werden. Daraufhin wird das neue Root-Verzeichnis festgelegt. Ein Prozess in diesem Container hat keinen Zugriff auf höhere Ordner mehr.



### 1.8.2.2 Namespaces

Mit Namespaces werden ausgewählte Bereiche in einem eigenen Container gruppiert. Es gibt Namespaces für verschiedene Aufgaben. Diese haben spezielle Namen.

Mit Mount (`mnt`) können Bereiche des Dateisystems gruppiert werden, wie im vorherigen Beispiel gezeigt wurde. Dies geschieht mit `unshare -m`.

Das Unix Time-share System (`uts`) ist dafür da, dass das gleiche System bei verschiedenen Prozessen unterschiedliche Host- und Domännennamen haben kann. Dies geschieht mit `unshare -u`.

Mit der Interprozess Kommunikation (`ipc`) können verschiedene Kommunikationskanäle zwischen den Prozessen eingerichtet werden.

Die Process-ID (`pid`) bildet einen Namespace bezüglich der Prozess-ID-Nummern und der Prozesssichtbarkeit. Dies geschieht mit `unshare -fp --mount-proc`.

Auch Netzwerke können Namespaces bilden. Hierzu werden Netzwerkinterfaces verwendet. Die Netzwerkeinrichtung erfolgt mit `ip`. Für die Abgrenzung wird `unshare -n` verwendet.

Über die User-ID (uid) können Namespaces gebildet werden. Dabei geht es um die Abgrenzung anhand von Benutzer-ID-Nummern und Benutzernamen mit unshare -U.

Eine weitere Möglichkeit bietet die Zeit (time). Die Systemzeit lässt sich mit unshare -T abgrenzen.

Weiterhin geht eine Abgrenzung der Syslogs, indem die Kernel Logs abgegrenzt werden.

Ein weiterer Namespace ist die Controll Group (cgroup). Diese ist für die Ressourcenmengenverwaltung zuständig und dessen Einstellungen befinden sich unter /sys/fs/cgroup.

Außerdem gibt es noch Composing Namespaces. Diese dienen dem Verknüpfen von Namespaces unterschiedlicher Kategorien. Mit dem Befehl nsenter wird ein Programm in verschiedenen Namespaces ausgeführt.

### 1.8.3 Docker

Docker bietet eine einfache Nutzung von Namespaces. Es stellt eine vordefinierte Softwareumgebung bereit, sodass die Installation einfach ist. In diesen Umgebungen lassen sich standardisierte Testumgebungen realisieren. Docker dient der Isolation von Services. Ein Docker-Container baut sich in Schichten wie Betriebssystem, Webserver, Konfigurationsdatei auf.

#### 1.8.3.1 Docker Begriffe

Im Folgenden werden einige Begriffe erläutert:

Layer = Befehl oder Datei

Image = Zusammenfassung mehrerer Layer

Dockerfile = Datei zur Beschreibung eines Images

Container = Instanz eines Images

Repository = Versionssatz eines Images

Registry = Zentraler Speicherort von Repositories

#### 1.8.3.2 Dockerfile Beispiel

Im Folgenden werden einige Beispiele dargestellt:

```
# Docker Image basiert auf nginx base image -> FROM nginx:latest
```

```
# Kopiere Konfiguration -> COPY nginx/nginx.conf /etc/nginx/nginx.conf
```

```
# Kopiere Bilder für Webserver -> CMD mkdir /var/www/images
```

```
COPY nginx/images /var/www/images
```

### 1.8.3.3 Docker Compose

Docker Compose ermöglicht es Komplexere Systeme über verknüpfte Docker Container aufzubauen. Dabei ist jeder Service ein eigener Container. Die Verknüpfung von Containern bildet eine Gesamtanwendung. Der Aufbau ist in `docker-compose.yml` beschrieben. Die folgende Abbildung zeigt als Beispiel ein Webserver mit verschiedenen Backend-Knoten und einer Datenbank:

```
webservers:
  build: ./nginx
  links:
    - node1:node1
    - node2:node2
  ports:
    - "80:80"

node1:
  build: ./node
  links:
    - database
  ports:
    - "8080"

node2:
  build: ./node
  links:
    - database
  ports:
    - "8080"

database:
  image: redis
  ports:
    - "6379"
```

### 1.8.3.4 Kubernetes (K8s)

Kubernetes ist für das Ausrollen von Containern zuständig. Es ist ausgerichtet auf Cluster (Computer-Pool). Die Ressourcen werden in einem Pool bereitgestellt. Der Nutzer selbst definiert Services, die er benötigt. Beispielsweise kann er sagen, dass er 5x Webserver, 2x Lagerverwaltungssoftware, 1x Fileshare-Server und 1x Honey Pod haben möchte und dies soll jetzt für ihn organisiert werden. Weiterhin reagiert Kubernetes auf einen Systemausfall.

## 1.9 Linux Sicherheit

### 1.9.1 Grundprinzipien der IT-Sicherheit

#### 1.9.1.1 Sicherheit Grundprinzipien

Aus der Praxis wird in Punkto Sicherheit nur so viel wie nötig eingesetzt, um die Angriffsfläche zu minimieren. Man nutzt eine Verteidigung in der Tiefe (defense in depth), um eine gestaffelte Sicherheitsstruktur und damit Redundanzen in der Sicherheit aufzubauen. Es ist wichtig für die Sicherheit, dass man seine Systeme und die Infrastruktur kennt. Dazu zählt was installiert ist und wer was nutzt. Wichtig ist, dass alles dokumentiert wird. Weiterhin sollte die Sicherheit immer überprüft werden, sprich ob es Unerwartetes gibt und welche Angriffsszenarien möglich sind.

In der Theorie gibt es das CIA-Prinzip. Hier geht es um den Schutz der drei Komponenten Vertraulichkeit (confidentiality), Integrität (integrity) und Verfügbarkeit (availability). Um die Vertraulichkeit zu schützen, dürfen nur Berechtigte Zugriff erlangen. Der Schutz vor unbemerkten Veränderungen sichert die Integrität und indem Ausfälle vermieden werden, wird die Verfügbarkeit geschützt. Weiterhin gibt es das AAA Prinzip. Hierbei geht es um Authentifizierung, Autorisierung und Accounting. Mit der Authentifizierung wird eine Benutzeridentifikation gewährleistet. Mit der Autorisierung wird geprüft, ob der Benutzer für etwas bestimmtes berechtigt ist und mit Accounting wird protokolliert, was dieser Benutzer tut.

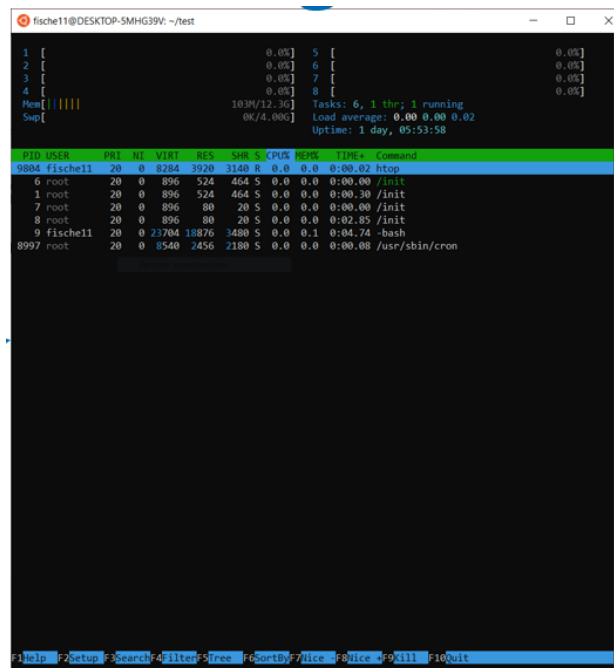
#### 1.9.1.2 Kerckhoffs' Prinzip (1883)

Die Geheimhaltung des Algorithmus, sprich Security by Obscurity (Sicherheit durch Verschleierung) ist eine schlechte Praxis. Ein neuer Algorithmus erfordert viel Arbeit und ist durch Reverse Engineering ermittelbar. Laut Kerckhoff ist die bessere Variante die Geheimhaltung des Schlüssels. Die Generierung eines Schlüssels ist einfach und ein schneller Austausch ist möglich. Somit ist es einfacher einen Schlüssel zu tauschen anstatt eines ganzen Algorithmus. Damit besagt das Kerckhoffsche Prinzip, dass die Sicherheit durch die Geheimhaltung des Schlüssels erlangt wird. Der Verschlüsselungsalgorithmus kann jedem bekannt sein. Dieses Prinzip ist der Grundsatz für die moderne Kryptographie.

## 1.9.2 Programmüberwachung

### 1.9.2.1 Laufende Programme anzeigen

Mit `ps -aux` kann man alle Prozesse anzeigen. Mit `pstree` wird die Startabhängigkeit bzw. die Startreihenfolge von Prozessen dargestellt. Der Befehl `top` entspricht dem „Taskmanager“. `htop` ist die verbesserte Version von `top`.



```
fische11@DESKTOP-5MHG39V: ~/test
1 [          0.0%]  5 [          0.0%]
2 [          0.0%]  6 [          0.0%]
3 [          0.0%]  7 [          0.0%]
4 [          0.0%]  8 [          0.0%]
Mem[|||||]          103M/12.3G    Tasks: 6, 1 thr; 1 running
Swap[         ]          0K/4.00G    Load average: 0.00 0.00 0.02
                                   Uptime: 1 day, 05:53:58

PID  PPID  PRI  NI  VSZ  RSS  SHR  %CPU  %MEM  TIME+  COMMAND
9804 fische11  20  0  8284 3920 3140 R  0.0  0.0  0:00.02 htop
6 root    20  0  896  524  464 S  0.0  0.0  0:00.00 /init
1 root    20  0  896  524  464 S  0.0  0.0  0:00.30 /init
7 root    20  0  896  80  20 S  0.0  0.0  0:00.00 /init
8 root    20  0  896  80  20 S  0.0  0.0  0:02.05 /init
9 fische11  20  0 23704 18876 3480 S  0.0  0.1  0:04.74 -bash
8997 root    20  0  8540 2456 2180 S  0.0  0.0  0:00.00 /usr/sbin/cron

F1 Help  F2 Setup  F3 Search  F4 Filter  F5 Tree  F6 SortBy  F7 Nice  F8 Nice  F9 Kill  F10 Quit
```

Die Systemüberwachung ist ebenfalls in der GUI möglich.

### 1.9.2.2 Programhash

Folgendes Angriffsszenario sei gegeben: `ps`, `top`, etc. wurde manipuliert und blendet Schadsoftware aus. Um eine Veränderung von Programmen zu erkennen kann `sha256sum <Programm>` verwendet werden, beispielsweise `sha256sum /bin/*`. Um regelmäßig den Hash zu prüfen, kann man ein Skript über `cron` nutzen. Zu beachten ist, dass auch ein Patch den Hashwert ändert.

```

fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ sha256sum /bin/ps
9badc3ea544cf1d5f3da165508291a28400773b3904749b7765fb2de61cff7db /bin/ps
fische11@DESKTOP-5MHG39V:~/test$ sha256sum /bin/pstree
8e43af67366ab733b795b8e4b8bfff265ef0b32c8e9f07cb333796e0baad44fe6 /bin/pstree
fische11@DESKTOP-5MHG39V:~/test$ sha256sum /bin/top
20110d06f2491dc37bd346fcfeebae81ce121dad9ff7f0c1b3ae05b4f9e0290d /bin/top
fische11@DESKTOP-5MHG39V:~/test$ sha256sum /bin/htop
1e14eb87ca42767242bb82f432dfc4cba3aa7d920323191cf943c368cc01b5bf /bin/htop
fische11@DESKTOP-5MHG39V:~/test$ sha256sum /bin/ls
1e39354a6e481dac48375bfebb126fd96aed4e23bab3c53ed6ecf1c5e4d5736d /bin/ls
fische11@DESKTOP-5MHG39V:~/test$ sha256sum /bin/cp
40ea53f38efe555c09a2c1c860379190e2af94427daaa68756874154c3fa1188 /bin/cp
fische11@DESKTOP-5MHG39V:~/test$ sha256sum /bin/apt
5fb01c6ec5839ad7c27af85b3c7dde8f7871d6d892ca52bd2e255b3e7a3f016a /bin/apt
fische11@DESKTOP-5MHG39V:~/test$ sha256sum /bin/apt-add-repository
4747a659dc4355edd0888f1825f086de5630ae56bd4f73d02e2c6096dfbb00bd /bin/apt-add-repository
fische11@DESKTOP-5MHG39V:~/test$ sha256sum /bin/sha256sum
b693d4bd026ba1515edacabcadb6785454b08ec9468665e4dc33f01df8cc659 /bin/sha256sum
fische11@DESKTOP-5MHG39V:~/test$

```

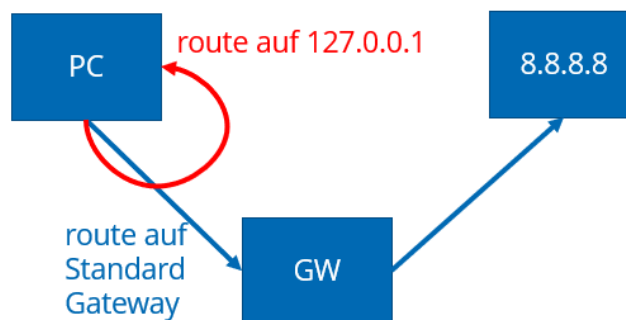
### 1.9.2.3 Updates

Updates sind die Sicherheitsempfehlung Nummer 1. Das Entfernen von Updates führt zu Fehlern und Sicherheitslücken. Automatisierte Updates können mit unattended-Upgrade oder mit cron-jobs durchgeführt werden. Weiterhin kann man Updates über den Paketmanager durchführen. Beispielsweise mit apt update, apt upgrade, apt dist-upgrade, apt autoremove und apt autoclean. Manuell installierte Programme sollte man auf die neuste Version prüfen und diese gegebenenfalls manuell aktualisieren.

## 1.9.3 Netzwerksicherheit

### 1.9.3.1 IP-Filter mit route

route kann als Ausgangsfilter genutzt werden. Hierbei gibt es eine feste Routingtabelle als Filter. Die Redundanz wird durch IP-Tables gewährleistet.



```
fische11@DESKTOP-5MHG39V: ~/test
fische11@DESKTOP-5MHG39V:~/test$ route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default DESKTOP-5MHG39V 0.0.0.0 UG 0 0 0 eth0
172.29.64.0 0.0.0.0 255.255.240.0 U 0 0 0 eth0
fische11@DESKTOP-5MHG39V:~/test$ ping -c 1 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=56 time=19.4 ms

--- 8.8.8.8 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 19.404/19.404/19.404/0.000 ms
fische11@DESKTOP-5MHG39V:~/test$ sudo route add -host 8.8.8.8 metric 10 dev lo
fische11@DESKTOP-5MHG39V:~/test$ route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default DESKTOP-5MHG39V 0.0.0.0 UG 0 0 0 eth0
dns.google 0.0.0.0 255.255.255.255 UH 10 0 0 lo
172.29.64.0 0.0.0.0 255.255.240.0 U 0 0 0 eth0
fische11@DESKTOP-5MHG39V:~/test$ ping -c 1 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.

--- 8.8.8.8 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms

fische11@DESKTOP-5MHG39V:~/test$ sudo route del 8.8.8.8
fische11@DESKTOP-5MHG39V:~/test$ route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default DESKTOP-5MHG39V 0.0.0.0 UG 0 0 0 eth0
172.29.64.0 0.0.0.0 255.255.240.0 U 0 0 0 eth0
fische11@DESKTOP-5MHG39V:~/test$ ping -c 1 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=56 time=24.7 ms

--- 8.8.8.8 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 24.714/24.714/24.714/0.000 ms
fische11@DESKTOP-5MHG39V:~/test$
```

### 1.9.3.2 DNS-Filter mit hosts

Die DNS-Auflösung ist in `/etc/nsswitch.conf` definiert. Standardmäßig befinden sich statische DNS-Einträge in der `/etc/hosts` Datei. Bei einer DNS-Auflösung schaut der Prozess zunächst in den lokalen Cache, dann in die Hosts Datei und schließlich kann DNS auch über den Server aufgelöst werden. Beispielweise sperrt ein Eintrag in der hosts-Datei: `127.0.0.1 zu_sperrende_Domain` den Zugang zu dieser Domainadresse. Zu beachten ist, dass die Subdomain nicht gesperrt wird. Auch Wildcards funktionieren nicht.



```

linuxmint@linuxmint-VirtualBox:~$ tail -n 2 /etc/hosts
127.0.0.1 google.at

linuxmint@linuxmint-VirtualBox:~$ nslookup google.at
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   google.at
Address: 127.0.0.1

linuxmint@linuxmint-VirtualBox:~$ nslookup google.de
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   google.de
Address: 142.250.185.131
Name:   google.de
Address: 2a00:1450:4001:803::2003

linuxmint@linuxmint-VirtualBox:~$ ping -c 1 google.at
PING google.at (127.0.0.1) 56(84) Bytes Daten:
64 Bytes von localhost (127.0.0.1): icmp_seq=1 ttl=64 Zeit=0.032 ms

--- google.at ping-Statistik ---
1 Pakete übertragen, 1 empfangen, 0% Paketverlust, Zeit 0ms
rtt min/avg/max/mdev = 0.032/0.032/0.032/0.000 ms
linuxmint@linuxmint-VirtualBox:~$ ping -c 1 google.de
PING google.de (142.250.185.131) 56(84) Bytes Daten:
64 Bytes von fra16s50-in-f3.1e100.net (142.250.185.131): icmp_seq=1 ttl=56 Zeit=21.3 ms

--- google.de ping-Statistik ---
1 Pakete übertragen, 1 empfangen, 0% Paketverlust, Zeit 0ms
rtt min/avg/max/mdev = 21.336/21.336/21.336/0.000 ms
linuxmint@linuxmint-VirtualBox:~$ nslookup www.google.at
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   www.google.at
Address: 216.58.212.163
Name:   www.google.at
Address: 2a00:1450:4001:802::2003

```

### 1.9.3.3 Offene Ports

Ein offener Port ist wie eine Öffnung im Haus. Nicht benötigte Ports sollten geschlossen werden. Offene Ports hingegen gilt es abzusichern. Dies kann man beispielsweise durch eine Authentifizierung machen. Weiterhin sollte man schauen welches Programm hinter dem Port ist und worauf das Programm Zugriff hat. netstat -tulpen zeigt die lauschenden Ports und aktiven Verbindungen an. nmap ist ein Netzwerksicherheitsscanner. Die Ausgabe für nmap -A 127.0.0.1 wird in der folgenden Abbildung gezeigt.

```

Linuxmint@linuxmint-VirtualBox:~$ sudo netstat -tulpen
Aktive Internetverbindungen (Nur Server)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       Benutzer   Inode      PID/Program name
tcp      0      0 127.0.0.1:631          0.0.0.0:*                LISTEN      0          23266      699/cupsd
tcp      0      0 127.0.0.53:53         0.0.0.0:*                LISTEN      101        18422     543/systemd-resolve
tcp      0      0 0.0.0.0:22           0.0.0.0:*                LISTEN      0          24715     739/sshd: /usr/sbin
tcp6     0      0 :::631                :::*                    LISTEN      0          23265      699/cupsd
tcp6     0      0 :::22                 :::*                    LISTEN      0          24717     739/sshd: /usr/sbin
udp      0      0 127.0.0.53:53         0.0.0.0:*                *          101        18421     543/systemd-resolve
udp      0      0 0.0.0.0:35100        0.0.0.0:*                *          119        23879     564/avahi-daemon: r
udp      0      0 0.0.0.0:631          0.0.0.0:*                *          0          24588     676/cups-browsed
udp      0      0 0.0.0.0:5353         0.0.0.0:*                *          119        23877     564/avahi-daemon: r
udp6     0      0 :::46186              :::*                    *          119        23880     564/avahi-daemon: r
udp6     0      0 :::5353               :::*                    *          119        23878     564/avahi-daemon: r

Linuxmint@linuxmint-VirtualBox:~$ sudo nmap -A 127.0.0.1
Starting Nmap 7.80 ( https://nmap.org ) at 2022-05-18 12:11 CEST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000079s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.4 (Ubuntu Linux; protocol 2.0)
631/tcp   open  ipp      CUPS 2.3
|_ http-robots.txt: 1 disallowed entry
|_/
|_ http-server-header: CUPS/2.3 IPP/2.1
|_ http-title: Home - CUPS 2.3.1
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 0 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.10 seconds
Linuxmint@linuxmint-VirtualBox:~$ sudo nmap -A -6 ::1
Starting Nmap 7.80 ( https://nmap.org ) at 2022-05-18 12:11 CEST
Nmap scan report for ip6-localhost (::1)
Host is up (0.0000080s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.4 (Ubuntu Linux; protocol 2.0)
631/tcp   open  ipp      CUPS 2.3
|_ http-server-header: CUPS/2.3 IPP/2.1
|_ http-title: Ung\xC3\xBCltige Anfrage - CUPS v2.3.1
Device type: general purpose
Running: Linux 3.Xl4.X

```

### 1.9.3.4 iptables

Mit iptables können Firewallregeln erstellt werden. Mit iptables -L -v können Regeln mit Details angezeigt werden. INPUT enthält Regeln für eingehende Pakete, FORWARD welche für weitergeleitete Pakete und OUTPUT für ausgehende Pakete. Mit ACCEPT wird etwas durchgelassen, mit DROP verworfen und mit REJECT ebenfalls verworfen, aber der Sender wird zusätzlich noch darüber informiert.

#### Regel hinzufügen

Um bei iptables eine Regel hinzuzufügen, gibt es mehrere Möglichkeiten. Es gibt:

- sudo iptables
  - -A <Ankunftsliste>
  - -i <Schnittstelle>
  - -p <Protokoll>
  - -s <Quelle>
  - --dport <Ziel-Port>
  - -j <Aktion>

- sudo iptables
  - -A INPUT
  - -i eth0
  - -p tcp
  - --dport 80
  - -j ACCEPT
- sudo iptables
  - -A FORWARD
  - -i tun0
  - -s 192.168.0.1
  - -j REJECT
- sudo iptables
  - -A INPUT
  - -i wlan0
  - -p tcp
  - --dport 22
  - -j ACCEPT

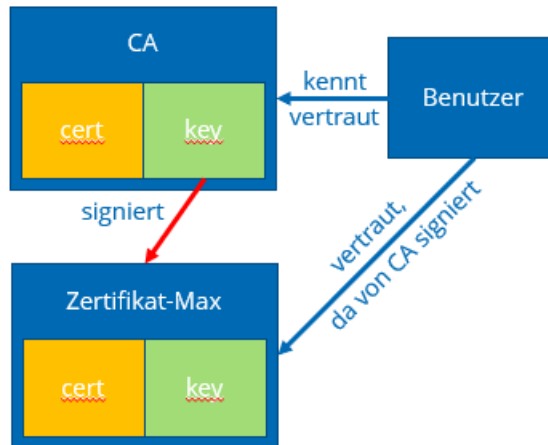
Eine Ankunftsliste findet man unter INPUT, FORWARD und OUTPUT. Die Schnittstelle befindet sich unter dem Netzwerkschnittstellennamen. Protokolle sind tcp, udp, icmp, oder Ähnliches. Die Quelle ist die Quell-IP, der Ziel-Port ist die Portnummer des Ziel-Portes. Mögliche Aktionen sind ACCEPT, DROP und REJECT.

### Regel löschen

Mit sudo iptables -L oder --line-numbers können Einträge nummeriert angezeigt werden. Man kann auch einen bestimmten Eintrag löschen. Dies geschieht mit dem Befehl sudo iptables -D <Ankunftsliste> <Eintragsnummer>. Beispielsweise wird mit sudo iptables -D INPUT 3 der 3. Eintrag der INPUT Liste gelöscht. Mit sudo iptables -F werden alle Einträge gelöscht. Änderungen erfolgen nur im RAM. Deshalb ist das Speichern für persistente Regeln notwendig, was mit sudo /sbin/iptables-save durchgeführt werden kann.

### 1.9.3.5 Zertifikate

Die Basis ist der X.509 Standard. Mit Zertifikaten erfolgt die Authentifizierung des Servers bei HTTPS, das Starten einer sicheren Verbindung über TLS, die VPN-Authentifizierung und die Signierung. Bei der Signierung werden PDF-Dateien, Programme und SSH-Keys signiert. Die oberste Einheit ist die CA (Certificate Authority). Die .crt ist das öffentliche Zertifikat mit dem öffentlichen Schlüssel. .key ist hierbei der private Schlüssel. Das Speicherformat ist PEM. Der Speicherort ist /etc/ca-certificates und /etc/ssl.



Die folgende Abbildung zeigt ein Zertifikat Beispiel:

ubuntusers.de		R3	ISRG Root X1
<b>Inhabername</b>			
Allgemeiner Name	ubuntusers.de		
<b>Ausstellername</b>			
Land	US		
Organisation	Let's Encrypt		
Allgemeiner Name	R3		
<b>Gültigkeit</b>			
Beginn	Sun, 01 May 2022 14:35:49 GMT		
Ende	Sat, 30 Jul 2022 14:35:48 GMT		

## Zertifikat erstellen

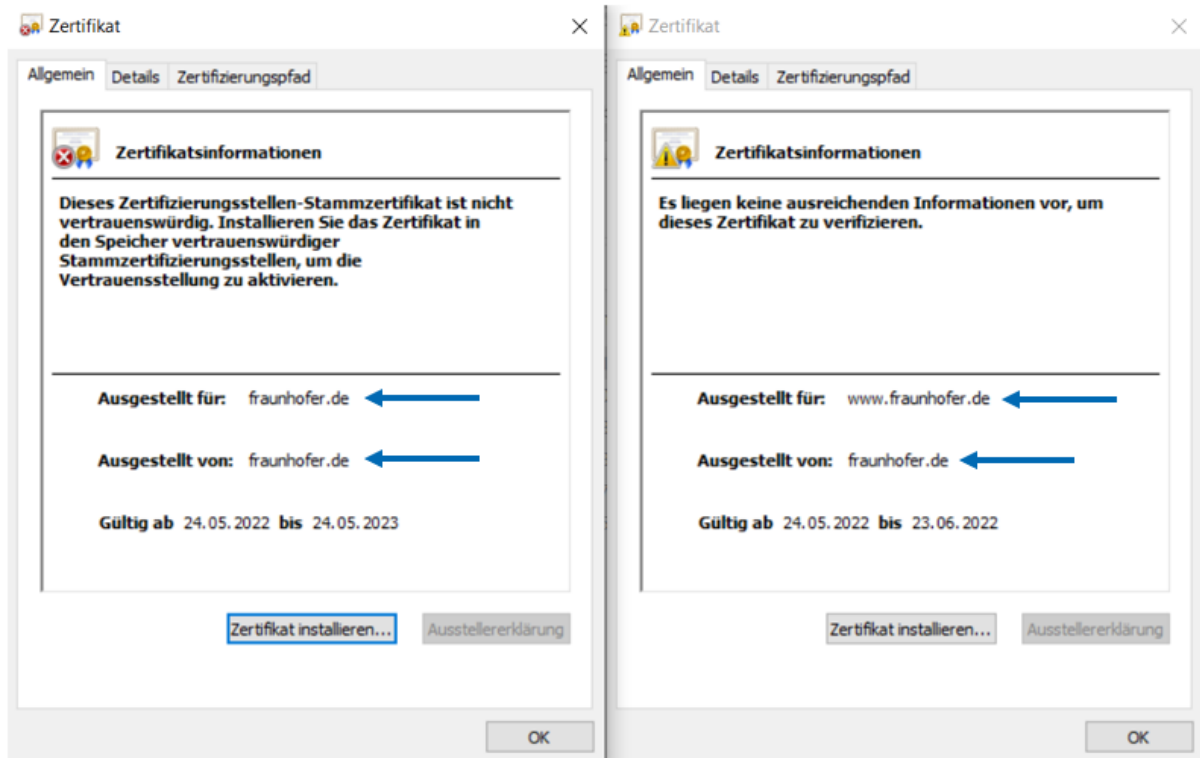
Man kann ein selbstsigniertes Zertifikat erstellen:

```
Fische11@DESKTOP-5PHG39V:~/ca$ openssl req -newkey rsa:4096 -x509 -sha512 -days 365 -nodes -out ca.crt -
keyout ca.key
Generating a RSA private key
.....++++
.....++++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:DE
State or Province Name (full name) [Some-State]:saxony
Locality Name (eg, city) []:Mittweida
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Fraunhofer e.V.
Organizational Unit Name (eg, section) []:Lernlabor Cybersicherheit
Common Name (e.g. server FQDN or YOUR name) []:fraunhofer.de
Email Address []:fische11@hs-mittweida.de
Fische11@DESKTOP-5PHG39V:~/ca$
```

Eine weitere Möglichkeit ist, dass Zertifikat zu erstellen und mit CA zu signieren:

```
Fische11@DESKTOP-5PHG39V:~/ca$ openssl genrsa -out webservers.key 4096
Generating RSA private key, 4096 bit long modulus (2 primes)
.....++++
.....++++
e is 65537 (0x010001)
Fische11@DESKTOP-5PHG39V:~/ca$ openssl req -new -key webservers.key -out webservers.csr -extensions v3_req
-sha512
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:DE
State or Province Name (full name) [Some-State]:saxony
Locality Name (eg, city) []:Mittweida
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Fraunhofer e.V.
Organizational Unit Name (eg, section) []:Lernlabor Cybersicherheit
Common Name (e.g. server FQDN or YOUR name) []:www.fraunhofer.de
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Fische11@DESKTOP-5PHG39V:~/ca$ openssl x509 -sha512 -req -in webservers.csr -out webservers.crt -CA ca.crt
-CAkey ca.key -days 30 -CAcreateserial -CAserial ca.seq
Signature ok
subject=C = DE, ST = saxony, L = Mittweida, O = Fraunhofer e.V., OU = Lernlabor Cybersicherheit, CN = ww
w.fraunhofer.de
Getting CA Private Key
Fische11@DESKTOP-5PHG39V:~/ca$ ls -l
total 24
-rw-r--r-- 1 fische11 fische11 2236 May 24 17:11 ca.crt
-rw----- 1 fische11 fische11 3272 May 24 17:10 ca.key
-rw-r--r-- 1 fische11 fische11 41 May 24 17:18 ca.seq
-rw-r--r-- 1 fische11 fische11 2061 May 24 17:18 webservers.crt
-rw-r--r-- 1 fische11 fische11 1748 May 24 17:17 webservers.csr
-rw----- 1 fische11 fische11 3243 May 24 17:15 webservers.key
```



### 1.9.3.6 USB-Sicherheit

Eine USB-Firewall lässt sich über `usbauth` aufbauen. Dabei kann man spezielle Geräte sperren, beispielsweise dass der USB-LTE-Adapter von Telecom nicht erlaubt wird. Weiterhin kann man Gerätegruppen sperren, wie WLAN-Adapter blockieren. Auch das Sperren an festen USB-Ports ist möglich. Hierbei wird beispielsweise die Tastatur nur an USB2.0 erlaubt. Ebenfalls ist es möglich, die Anzahl zu beschränken, dass nur eine Tastatur erlaubt ist.

Im Folgenden werden einige Beispiele gezeigt:

```
allow busnum==3 devpath==2 bInterfaceClass==03 anyChild bInterfaceProtocol==01 devcount<=1
```

➔ Hierbei wird nur eine Tastatur an BUS 3, Anschluss 2 erlaubt

```
allow busnum==2 devpath==6 idVendor==1c7a idProduct==0801
```

➔ Hierbei wird ein spezieller Fingerabdrucksensor an BUS 2, Anschluss 6 erlaubt

```
deny all
```

➔ Alles andere wird verboten

## 1.9.4 Rechteverwaltung

### 1.9.4.1 Datei-Zugriffsberechtigung

Die Datei-Zugriffsberechtigung wird in folgendem Format dargestellt: `drwxrwxrwx` (TypBesitzerGruppeAlleAnderen). `d` steht für das Dateiformat. Folgende stehen zur Verfügung:

- -: normale Datei
- d: Ordner (directory)
- b: Block-Device
- c: Zeichen (character)
- l: Softlink (symbolic link)
- n: Netzwerk (network)
- p: Pipe / First-in, First-out
- s: Socket

`r` sind die Lese-, `w` die Schreib- und `x` die Ausführungsrechte. `s` steht für `setuid`, sprich die Ausführung mit Benutzerrechten, `S` für `setgid`, sprich die Ausführung mit Gruppenrechten und `t` ist das `stickybit`, sprich das Programm verbleibt nach dem Ausführungsende im Speicher.

### 1.9.4.2 Ordnerrechte

`R` steht für Leserechte. Hiermit ist das Auflisten der Dateien im Ordner möglich. Mit `W` hat man Schreibrechte, womit das Anlegen, Bearbeiten und Löschen von Dateien möglich ist. `X` erlaubt den Zugriff, wobei man Zugriff auf die Dateien im Ordner hat, wenn der Name bekannt ist.

### 1.9.4.3 User-Typen

Es gibt verschiedene User-Typen. Zum einen gibt es den Normal-User. Dieser hat ein Benutzerkonto, welches für die normale Anmeldung gedacht ist. Weiterhin hat er ein Home-Verzeichnis für persönliche Einstellungen und private Dateien. Bei den Normal-Usern gibt es noch den speziellen `root` Nutzer. Dieser hat Vollzugriff und darf somit alles. Er hat die `UID 0` und ist ein attraktives Ziel für Angreifer. Normal-User haben eine `UID` ab `1000`.

Zum anderen gibt es den System-User. Dies ist ein Benutzer für Programme und für die Rechteverwaltung von Programmen. Dabei wird das Programm unter dem System-User-Kontext gestartet. Die Benutzer-Rechte bestimmen hierbei die Programm-Rechte. Der System-User hat kein Home-Verzeichnis und wird meist automatisch bei der Programminstallation erstellt.

### 1.9.4.4 Gruppen-Typen

Es gibt eine normale-Gruppen und eine System-Gruppe. Die normale-Gruppen sind die Gruppen für Benutzer, wobei die `GID` ab `1000` beginnt. Die `Root-Gruppe` hat die `GID 0`. Die normalen-Gruppen



werden standardmäßig bei der Benutzererstellung erstellt. Die Gruppe wird nach dem Benutzer benannt und erhält dieselbe ID.

Die System-Gruppe ist die Gruppe für System-Nutzer. Es gibt keinen Unterschied für die Rechte zu den normalen Gruppen. Sie werden bei Anzeigen von Benutzern ausgeblendet (optionale Anzeige).

#### 1.9.4.5 Passwortsicherheit

Das Passwort wird nicht im Klartext gespeichert. Für die Passwortabfrage wird ein Passwort-Hash benutzt. Der Salt schützt vor Rainbow-Table Angriffen. Zu beachten ist, dass die Länge die Zeichenanzahl schlägt:

$$\text{Dauer} = \frac{\text{Zeichenanzahl}^{\text{Länge}}}{\text{Versuche pro Sekunden}}$$

Tipp: Man sollte 3-4 Wörter mit Sonderzeichen/Zahlen dazwischen verwenden.

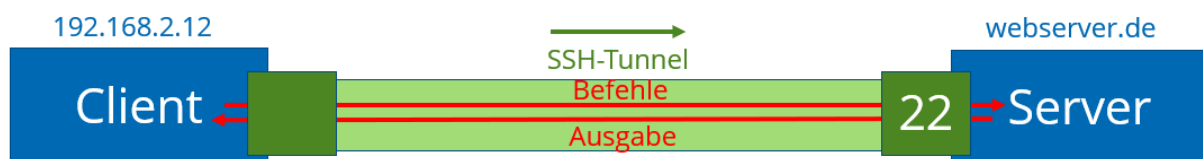
#### 1.9.4.6 Benutzer sperren

Um einen Benutzer zu sperren muss man den Eintrag in /etc/shadow ändern: Benutzer:\*:.... Ein !, !! oder \* verhindert den Login und ersetzt dabei den Passwort-Salt und -Hash.

### 1.9.5 Fernzugriff

#### 1.9.5.1 SSH

SSH steht für Secure Shell und ist für eine sichere Verbindung zu entfernten Computern zuständig, welche verschlüsselt und authentifiziert ist. Sie kann automatisiert durch Skripte genutzt werden. SSH bietet eine sichere Datenübertragung und läuft standardmäßig auf Port 22. SSH ist zusätzlich als VPN oder Graphische Aufschaltung nutzbar.



## SSH-Login

Für den SSH-Login nutzt man „ssh benutzer@ip\_oder\_dns“. Bei der ersten Anmeldung muss der Fingerabdruck des Servers bestätigt werden, welcher in `~/.ssh/known_hosts` hinterlegt ist. Eine Änderung des Fingerabdrucks bedeutet, dass der Server neu installiert wurde oder einen “Man in the Middle Angriff” (Client <-> Angreifer <-> Server). Mit `-p` kann man einen von 22 abweichenden Port angeben.

## SSH-Key

Die Anmeldung erfolgt über das Public-Key-Verfahren. Hierbei ist keine Passwort Eingabe nötig. Als Kryptographieverfahren stehen RSA (Primfaktoren), DAS (Diskreter Logarithmus) und ECDSA und EDDSA (Elliptische Kurven Diskreter Logarithmus) zur Verfügung. Man hat ein Key-Pair pro Ziel-Computer. Die erlaubten Public-Keys findet man in `~/.ssh/authorized_keys` (Eine Zeile = Ein Eintrag). Die eigenen Keys hingegen befinden sich in `~/.ssh/key_name`, wobei dieser Private Key nie den Computer verlassen darf und immer unlesbar für andere Benutzer sein muss. Dagegen kann der öffentliche Schlüssel allen mitgeteilt werden. Dieser befindet sich unter `~/.ssh/key_name.pub` und wird auf die entsprechenden Server kopiert.

## SSH-Key einrichten

Zuerst wird der Schlüssel erzeugt. Dies geschieht mit `ssh-keygen -t ed25519 -a 420 -f ~/.ssh/server_name -C „Felix Laptop“`. `-t` kennzeichnet den Schlüsseltyp (ed25519 = Elliptische Kurve), `-a` die Hashrunden für den privaten Schlüssel (Brute Force Schutz), `-f` den Dateinamen und `-C` den Kommentar, welcher der Identifikation des Schlüssels auf dem Server dient.

Als nächstes wird der Schlüssel auf den Server übertragen. Dies kann alternativ manuell mit `scp` erfolgen. Der Befehl für die Übertragung ist `ssh-copy-id -i ~/.ssh/server_name.pub benutzer@server_ip_oder_dns`.

Zuletzt erfolgt die Verbindung mit der Nutzung des SSH-Keys: `ssh -i ~/.ssh/server_name.pub benutzer@server_ip_oder_dns`.

## SSH-Config

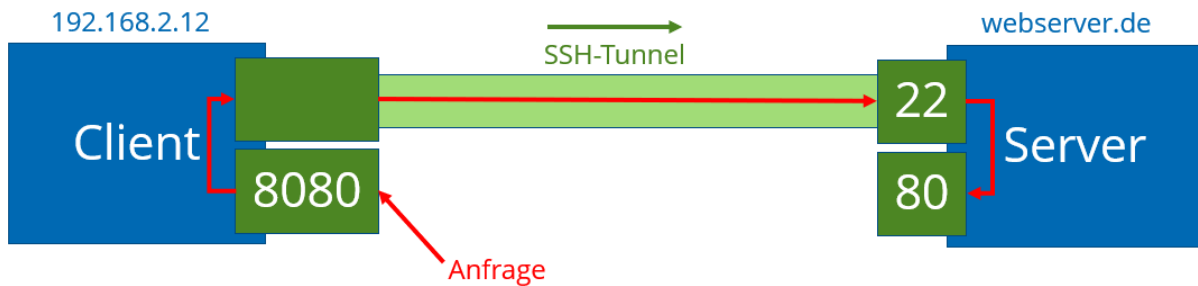
Die Konfiguration für die SSH-Verbindung wird unter `~/.ssh/config` gespeichert. Bei manchen Distributionen auch unter `~/.ssh/config.d/*`. Die Konfiguration sieht wie folgt aus:

- Host server\_name
  - HostName ssh.example.com
  - User benutzername
  - Port 22
  - PreferredAuthentications publickey
  - IdentityFile ~/.ssh/private\_ssh\_key
  - IdentitiesOnly yes

## SSH-Tunnel Forward

Der SSH-Tunnel Forward wird wie folgt aufgebaut: `ssh -L <localerPort>:eigeneIP:<serverPort> benutzer@server_ip`.

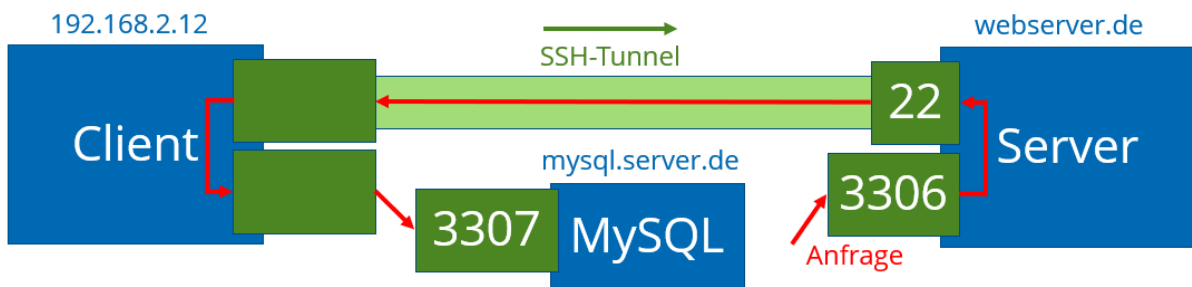
Ein Beispiel wäre: `ssh -L 8080:192.168.2.12:80 felix@webserver.de`. Damit würde eine SSH-Verbindung mit `webserver.de` aufgebaut werden. Der Port 80 wäre vom Server unter `192.168.2.12:80` erreichbar.



## SSH-Tunnel reverse

Ein SSH-Tunnel wird reverse wie folgt aufgebaut: `ssh -R <localerPort>:localeIP:<serverPort> benutzer@server_ip`.

Ein Beispiel könnte wie folgt aussehen: `ssh -R 3307:mysql.server.de:3306 felix@webserver.de`. Hierbei würde eine SSH-Verbindung mit `webserver.de` aufgebaut werden. Der Port 3307 ist vom MySQL-Server erreichbar und vom Webserver unter 3306.



### 1.9.5.2 SCP

scp steht für secure copy und führt das Kopieren durch einen SSH-Tunnel durch. Der Befehl lautet wie folgt: scp datei kopie (kein SSH-Tunnel). Weitere Beispiele wären:

- scp datei benutzer@server:/pfad/dateiname
- scp benutzer@server:/pfad/dateiname datei
- scp benutzer@server:/pfad/dateiname benutzer@server:/pfad/dateiname

scp unterstützt die Autovervollständigung. Eine Angabe vom Alias ist möglich, wenn es in ~/.ssh/config definiert ist.

### 1.9.6 Dateisicherheit

#### 1.9.6.1 GnuPG (Version2)

GnuPG ist für die Dateiverschlüsselung zuständig. Unterstützte Algorithmen sind: RSA, ECDH, ECDSA und EdDSA, Elgamal, DAS, AES, Camellia, Twofish und SHA2 (SHA256, SHA512).

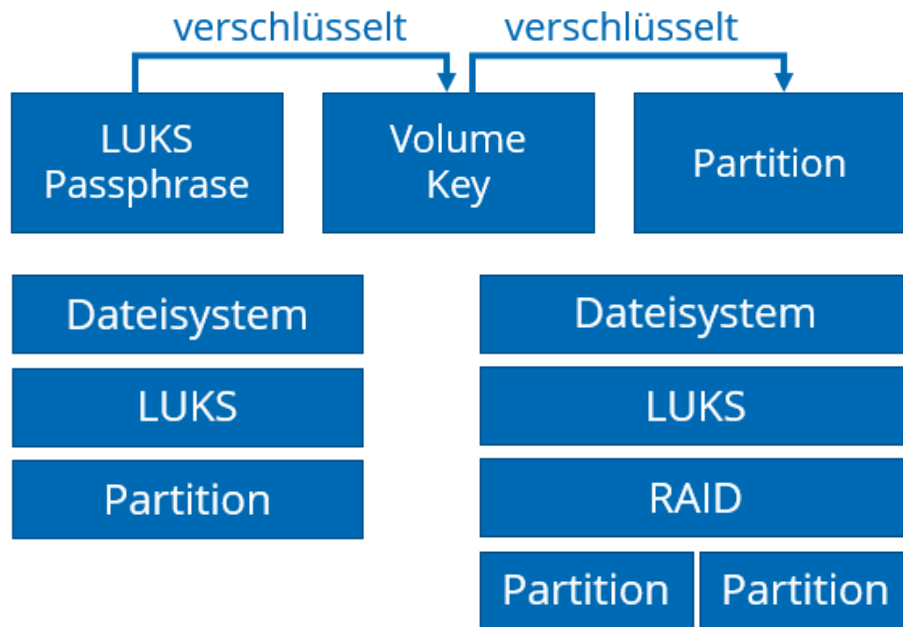
GnuPG Version 2 wird mit apt install gnupg2 installiert.

```
fische11@DESKTOP-5MHG39V:~/test$ cat file1
Das ist die Datei file1.
Nur ein wenig Text.
Eine weitere Zeile.
fische11@DESKTOP-5MHG39V:~/test$ gpg2 -c file1
gpg: directory '/home/fische11/.gnupg' created
gpg: keybox '/home/fische11/.gnupg/pubring.kbx' created
fische11@DESKTOP-5MHG39V:~/test$
fische11@DESKTOP-5MHG39V:~/test$ cat file1.gpg
    54Tv k^    jj=SF38! ;Z Z=@YL"<}~)}Yc ('>^oca,;h 4 EK5K >o -? x~Zr fische11@DESKTOP-5MHG39V
~/test$
fische11@DESKTOP-5MHG39V:~/test$ gpg -d -o file1.decrypted file1.gpg
gpg: AES256 encrypted data
gpg: encrypted with 1 passphrase
fische11@DESKTOP-5MHG39V:~/test$ cat file1.decrypted
Das ist die Datei file1.
Nur ein wenig Text.
Eine weitere Zeile.
fische11@DESKTOP-5MHG39V:~/test$ hexdump ~/.gnupg/pubring.kbx
00000000 0000 2000 0101 0200 424b 6658 0000 0000
00000100 8c62 6d9b 8c62 6d9b 0000 0000 0000 0000
00000200
fische11@DESKTOP-5MHG39V:~/test$
```

Aber GnuPG2 sollte nicht mit PGP verwechselt werden!

### 1.9.6.2 LUKS

LUKS steht für Linux Unified Key Setup und es handelt sich um eine Festplattenverschlüsselung (Partitionen oder Container). LUKS speichert Metadaten über die Verschlüsselungsmethode ab.



Für LUKS kann das Programm cryptsetup verwendet werden. Automatisiertes Einbinden ist hier möglich. Außerdem ist es in Dateibrowsern integriert.

```
fische11@DESKTOP-5MHG39V:~/test$ sudo cryptsetup luksFormat --type luks2 /dev/loop1
WARNING!
=====
This will overwrite data on /dev/loop1 irrevocably.

Are you sure? (Type uppercase yes): YES
Enter passphrase for /home/fische11/test/disk1:
Verify passphrase:
WARNING: Locking directory /run/cryptsetup is missing!
fische11@DESKTOP-5MHG39V:~/test$ sudo cryptsetup luksOpen /dev/loop1 loop1-decrypte
Enter passphrase for /home/fische11/test/disk1:
fische11@DESKTOP-5MHG39V:~/test$ sudo mkfs.ext4 /dev/mapper/loop1-decrypte
mke2fs 1.45.5 (07-Jan-2020)
Creating filesystem with 4096 4k blocks and 4096 inodes

Allocating group tables: done
Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done

fische11@DESKTOP-5MHG39V:~/test$ mkdir loop1-mount
fische11@DESKTOP-5MHG39V:~/test$ sudo mount /dev/mapper/loop1-decrypte loop1-mount
```

### 1.9.6.3 Sicheres Löschen

Der Befehl `rm` entfernt nur einen Eintrag im Filesystem. Dies würde in etwa dem Radieren im Inhaltsverzeichnis eines Buches entsprechen. Die Daten sind noch vorhanden, aber als überschreibbar markiert. Das vollständige Löschen ist das Überschreiben. Sicheres Löschen hingegen ist das mehrfache Überschreiben. Dies kann mit `shred -n 5 -uzv /Pfad/Datei` durchgeführt werden. `-n` ist die Anzahl an Überschreibungen, `-u` bedeutet, dass die Datei gelöscht und noch überschrieben wird. `-z` bedeutet, dass die Datei beim letzten Durchlauf mit Nullen überschrieben wird und `-v` zeigt Fortschrittsinformationen an. Eine weitere Möglichkeit ist `dd if=/dev/urandom of=/dev/Festplatte bs=4k`.

### 1.9.7 Zusammenfassung

Sie kennen nun Grundprinzipien der IT-Sicherheit.

Unter Linux haben Sie heute Möglichkeiten kennengelernt, wie Sie ein laufendes System auf auffällige Programme untersuchen können. Dazu zählen Programme, wie `ps`, `top` und `htop`. Außerdem wissen Sie, wie deren Integrität geprüft werden kann.

Sie können die Netzwerksicherheit eines Systems beurteilen und eigenständig Regeln für dieses festlegen. Sichere Datenübertragung von Befehlen und Daten können Sie mit SSH und SCP realisieren.

Sie wissen nun den Unterschied zwischen normalen und System-Benutzern bzw. Gruppen. Sie wissen außerdem, wie ein Benutzer gesperrt werden kann.

Heute haben Sie gelernt Daten und Festplatten unter Linux zu verschlüsseln und sicher zu löschen. Sie kennen den Unterschied zwischen einfachem Löschen und sicheren Löschen.