



# LINUX GRUNDKURS SKRIPTUM

# INHALT

<b>Grundlagen .....</b>	<b>4</b>
Historie .....	5
Der Kernel .....	5
Einsatzbereich.....	5
<i>Desktop-Systeme</i> .....	6
<i>Server</i> .....	7
<i>Smartphones und Tablets</i> .....	7
Systemstart .....	8
<i>Abgrenzung BIOS/ UEFI</i> .....	8
Grafische Benutzeroberfläche .....	8
<i>Die Basis: X-Window-System</i> .....	9
<i>KDE: Der Konsistente</i> .....	9
<i>Gnome: Der Desktop-Riese</i> .....	9
<i>Xfce: Der Tugendhafte</i> .....	10
Hilfreiche Links .....	10
<b>Dateiverwaltung .....</b>	<b>11</b>
Datenträgerverwaltung .....	11
Das Dateisystem .....	12
<i>Btrfs-Dateisystem</i> .....	12
<i>EXT-Dateisystem</i> .....	13
<i>Weitere</i> .....	13
Die Verzeichnisstruktur .....	13
<i>/ (Root-Verzeichnis)</i> .....	14
<i>/bin</i> .....	14
<i>/boot</i> .....	14
<i>/dev</i> .....	14
<i>/etc</i> .....	14
<i>/home</i> .....	15
<i>/lib</i> .....	15
<i>/lost+found</i> .....	15
<i>/mnt</i> .....	15
<i>/opt</i> .....	16
<i>/root</i> .....	16
<i>/usr</i> .....	16

<i>/var</i> .....	16
Nützliche Tools .....	17
<b>Rechteverwaltung</b> .....	<b>18</b>
sudo & root – Konzept der Rechte .....	18
Benutzer und Gruppen .....	18
Dateisystemrechte .....	19
<i>Darstellungsarten</i> .....	20
<i>Sonderrechte</i> .....	21
Nützliche Tools .....	21
<b>Paketverwaltung</b> .....	<b>22</b>
Konzept Anhand „Ubuntu“ .....	22
Weitere Paketverwaltungen .....	23
Pakete & Paketquellen .....	23
Nützliche Tools .....	24
<b>Textbasierte Anwendung</b> .....	<b>25</b>
Grundlagen .....	25
<i>Auflisten</i> .....	25
<i>Ausgeben</i> .....	25
<i>Hilfe anzeigen</i> .....	25
<i>Wechseln zwischen Verzeichnissen</i> .....	25
<i>Sonstiges</i> .....	26
Dateimanagement .....	26
<i>Anlegen von Verzeichnissen/Dateien</i> .....	26
<i>Löschen von Verzeichnissen/Dateien</i> .....	26
<i>Anzeigen von Verzeichnissen/Dateien</i> .....	26
<i>Kopieren von Verzeichnissen/Dateien</i> .....	26
<i>Verschieben von Verzeichnissen/Dateien</i> .....	26
<i>Softlink</i> .....	27
<i>Hardlink</i> .....	27
Suchen und Finden .....	27
<i>find</i> .....	27
<i>locate</i> .....	27
<i>egrep</i> .....	27
<i>Suchen nach Programmen</i> .....	28
Rechtemanagement .....	28
<i>Anzeigen von Rechten</i> .....	28

<i>Benutzer wechseln</i> .....	28
<i>User-Rechte</i> .....	28
<i>Group-Rechte</i> .....	28
<i>Others-Rechte</i> .....	28
<i>Administratoren-Rechte</i> .....	28
Mounten .....	28
<i>Manuelles mounten</i> .....	28
Unmount.....	29
Prozessverwaltung.....	29
<i>Herausfinden der Prozess-ID</i> .....	29
<i>Vorder- und Hintergrundausführung</i> .....	29
<i>Managen von Prozessen</i> .....	29
<i>Überwachung von Prozessen</i> .....	30
<i>Beenden von Prozessen</i> .....	30
<b>Best Practice</b> .....	<b>31</b>
Informationen anzeigen .....	31
<i>Hilfe anzeigen</i> .....	31
<i>Informationen zum Betriebssysteme, etc.</i> .....	31
.bashrc .....	32
<i>colorful terminal</i> .....	32
<i>alias für Befehle</i> .....	32
Updates automatisch Installieren (crontab) .....	33
<i>crontab anlegen</i> .....	33
Byobu.....	34
Dateimanagement .....	34
<i>Schnelles erstellen</i> .....	34
<i>Schnelles umbenennen</i> .....	34
<i>secure copy</i> .....	34
Pakteinstallation .....	35
vim .....	35
<i>Bedienung</i> .....	35
<b>Troubleshooting</b> .....	<b>37</b>
„Taskmanager“ .....	37
Log-Files .....	37
Dienst neu starten .....	38
Netzwerk.....	39

# SKRIPTUM

Dieses Skriptum stellt eine schriftliche Unterstützung zur Weiterbildung „LINUX 101“ dar, ist aber nicht als Ersatz für diese zu sehen. In der Weiterbildung werden alle im Skriptum vorkommenden Themen behandelt, mit dem Unterschied, dass im Skriptum mehr Theorie zu den Themen vorhanden ist, die als „nice to have“ anzusehen ist, aber nicht essentiell für den guten Umgang mit LINUX ist.

Dieses Skriptum wurde von der StV Lehramt der TU Graz durch die Leitung der Weiterbildung mit größter Sorgfalt erstellt, kann aber trotzdem Tippfehler o.ä. enthalten. Wir versuchen das Skriptum natürlich immer „state of the art“ zu halten.

Für die Erstellung des Skriptums sind verschiedene Quellen verwendet worden. Diese sind jeweils bei den Absätzen oder den Kapiteln, in denen sie verwendet wurden, ausgewiesen mit einem Verweis.

Alle Befehle, welche am Terminal ausgeführt werden können, werden in folgendem Format dargestellt:

```
| echo „hello world...“ >> linux101.txt
```

In diesem Sinne „per aspera ad astra“!

## GRUNDLAGEN

Als Linux bezeichnet man in der Regel freie, unix-ähnliche Mehrbenutzer-Betriebssysteme, die auf dem Linux-Kernel und wesentlich auf GNU-Software basieren. Die weite, auch kommerzielle Verbreitung wurde ab 1992 durch die Lizenzierung des Linux-Kernels unter *der freien Lizenz GPL*<sup>1</sup> ermöglicht. Einer der Initiatoren von Linux war der finnische Programmierer *Linus Torvalds*<sup>2</sup>. Er nimmt bis heute eine koordinierende Rolle bei der Weiterentwicklung des Linux-Kernels ein und wird auch als Benevolent Dictator for Life (deutsch wohlwollender Diktator auf Lebenszeit) bezeichnet.

Das modular aufgebaute Betriebssystem wird von Softwareentwicklern auf der ganzen Welt weiterentwickelt, die an den verschiedenen Projekten mitarbeiten. An der Entwicklung sind Unternehmen, Non-Profit-Organisationen und viele Freiwillige beteiligt. Beim Gebrauch auf Computern kommen meist sogenannte Linux-Distributionen zum Einsatz. Eine Distribution fasst den Linux-Kernel mit verschiedener Software zu einem Betriebssystem zusammen, das für die Endnutzung geeignet ist. Dabei passen viele Distributoren und versierte Benutzer den Kernel an ihre eigenen Zwecke an.

Linux wird vielfältig und umfassend eingesetzt, beispielsweise auf Arbeitsplatzrechnern, Servern, Mobiltelefonen, Routern, Netbooks, Embedded Systems, Multimedia-Endgeräten und Supercomputern. Dabei wird Linux unterschiedlich häufig genutzt: So ist Linux im Server-Markt wie auch im mobilen Bereich eine feste Größe, während es auf dem Desktop und Laptops eine noch geringe, aber wachsende Rolle spielt.

---

<sup>1</sup> Die GNU General Public License (kurz GNU GPL oder GPL; aus dem Englischen wörtlich für allgemeine Veröffentlichungserlaubnis oder -genehmigung) ist die am weitesten verbreitete Softwarelizenz, die einem gewährt, die Software auszuführen, zu studieren, zu ändern und zu verbreiten (kopieren).

<sup>2</sup> Linus Benedict Torvald (\* 28. Dezember 1969 in Helsinki) ist ein finnisch-US-amerikanischer Informatiker. Torvalds ist der Initiator sowie die treibende Kraft bei der Entwicklung des Linux-Kernels, dessen Entwicklung er bis heute koordiniert. Außerdem ist er der Erfinder und langjähriger Entwickler des Versionsverwaltungssystems Git.

Linux wird von zahlreichen Nutzern verwendet, darunter private Nutzer, Regierungen und Organisationen wie das Französische Parlament und das US-Verteidigungsministerium, Unternehmen wie Samsung, Siemens, Google, Amazon, Peugeot usw.<sup>3</sup>

## HISTORIE<sup>4</sup>

1991 begann Linus Torvalds in Helsinki (Finnland) mit der Entwicklung einer Terminal-Emulation, um unter anderem seinen eigenen Computer besser zu verstehen. Mit der Zeit merkte er, dass sich das System immer mehr zu einem Betriebssystem entwickelte; daraufhin kündigte er es in der Usenet-Themengruppe für das Betriebssystem *Minix*<sup>5</sup>, comp.os.minix, an. Im September desselben Jahres sollte das System dann auf einem Server den Interessierten zur Verfügung gestellt werden. Dem damaligen FTP-Server-Administrator Ari Lemmke gefiel keiner der von Torvalds vorgeschlagenen Namen Freax oder Buggix, deshalb veröffentlichte er es stattdessen in einem Verzeichnis mit dem Namen Linux. Torvalds war mit diesem Namen zunächst nicht einverstanden, gab seinen Widerstand aber schnell auf, weil er nach eigener Aussage eingestehen musste, dass Linux einfach ein besserer Name war.

Linux wurde zu dieser Zeit noch unter einer proprietären Lizenz von Torvalds veröffentlicht, die die kommerzielle Nutzung verbot. Er merkte jedoch bald, dass das den Fortschritt der Entwicklung behinderte. Er wollte allen Entwicklern deutlich mehr Freiraum geben und stellte Linux deshalb im Januar 1992 unter die GNU GPL. Es war nun möglich, Linux in GNU zu integrieren und dies als das erste freie Betriebssystem zu vertreiben. Dieser Schritt machte das System für eine noch größere Zahl von Entwicklern interessanter, da er die Modifizierung und Verbreitung vereinfachte.

## DER KERNEL<sup>6</sup>

Die Bezeichnung Linux wurde von Linus Torvalds anfänglich nur für den Kernel genutzt, dieser stellt der Software eine Schnittstelle zur Verfügung, mit der sie auf die Hardware zugreifen kann, ohne sie genauer zu kennen. Der Linux-Kernel ist ein in der Programmiersprache C geschriebener monolithischer Kernel. Wichtige Teilroutinen sowie zeitkritische Module sind jedoch in prozessorspezifischer Assemblersprache programmiert. Der Kernel ermöglicht es, nur die für die jeweilige Hardware nötigen Treiber zu laden. Weiterhin übernimmt der Kernel auch die Zuweisung von Prozessorzeit und Ressourcen zu den einzelnen Programmen, die auf ihm gestartet werden. Bei den einzelnen technischen Vorgängen orientiert sich das Design von Linux stark an seinem Vorbild Unix.

Der Linux-Kernel wurde zwischenzeitlich auf eine sehr große Anzahl von Hardware-Architekturen portiert. Das Repertoire reicht von eher exotischen Betriebsumgebungen wie dem iPAQ-Handheld-Computer, Navigationsgeräten von TomTom oder gar Digitalkameras bis hin zu Großrechnern wie IBMs System z und neuerdings auch Mobiltelefonen wie dem Motorola A780 sowie Smartphones mit Betriebssystemen wie Android oder Sailfish OS auf dem Jolla. Trotz Modulkonzept blieb die monolithische Grundarchitektur erhalten. Die Orientierung der Urversion auf die verbreiteten x86-PCs führte früh dazu, verschiedenste Hardware effizient zu unterstützen und die Bereitstellung von Treibern auch unerfahrenen Programmierern zu ermöglichen. Die hervorgebrachten Grundstrukturen beflügelten die Verbreitung

## EINSATZBEREICH<sup>7</sup>

---

<sup>3</sup> Vgl. <https://de.wikipedia.org/wiki/Linux>, aufgerufen am 25.12.2017

<sup>4</sup> Vgl. <https://de.wikipedia.org/wiki/Linux>, aufgerufen am 25.12.2017

<sup>5</sup> Minix (Eigenschreibweise MINIX) ist ein freies unixoides Betriebssystem, das von Andrew S. Tanenbaum an der Freien Universität Amsterdam als Lehrsystem entwickelt wurde.

<sup>6</sup> Vgl. <https://de.wikipedia.org/wiki/Linux>, aufgerufen am 25.12.2017

<sup>7</sup> Vgl. <https://de.wikipedia.org/wiki/Linux>, aufgerufen am 25.12.2017

---

## DESKTOP-SYSTEME

Linux, beziehungsweise eine Linux-Distribution, lässt sich als allein installiertes Betriebssystem betreiben, aber auch innerhalb eines Multi-Boot-Systems einsetzen. Parallel installieren kann man Linux beispielsweise neben Microsoft Windows oder einem BSD wie FreeBSD oder macOS. Moderne Distributionen wie OpenSUSE, Debian oder Ubuntu führen den Nutzer mit Hilfe von grafischen Benutzeroberflächen durch die Installation auf dem PC und erkennen andere Betriebssysteme nahezu immer selbstständig. Aus weit über tausend kostenlosen Programmen kann eine individuelle Kombination ausgewählt werden. Textverarbeitung, Tabellenkalkulation, Multimedia-Anwendungen, Netzwerktools, Spiele oder wissenschaftliche Anwendungen decken die meisten Anwendungsbereiche ab, die im Büroalltag und im Privatbereich wichtig sind.

Trotz Sicherheitsvorsprungs gegenüber dem am weitesten verbreiteten Betriebssystem Windows und der Möglichkeit der Parallelinstallation und umfangreichen, kostenlosen Softwareangebots wird Linux auf Desktoprechnern zögerlich eingesetzt. Auch wenn sich die verbreitetsten Linux-Desktop-Umgebungen ähnlich bedienen lassen wie Windows oder macOS, unterscheiden sie sich durch diverse Systemfunktionen von ihnen. Daher kann wie bei fast jedem Wechsel des Betriebssystems eine gewisse Einarbeitungszeit nötig sein. Im Gegensatz zur geringen Verbreitung auf dem Desktop ist Linux auf Server-Systemen, bei Embedded-Systemen und auf Smartphones bereits ein etabliertes Betriebssystem.

Die Installation der meisten Distributionen ist einfach und gibt geläufige Einstellungen vor, auch die Installation der Anwendungen läuft meist vollautomatisch ab, da sie üblicherweise von einem Paketmanager übernommen wird. Da das genaue Vorgehen aber nicht bei allen Linux-Distributionen einheitlich geregelt ist, kann ein Wechsel der Linux-Distribution Einarbeitungszeit erfordern. Die Installation von Programmen, die nicht zum Umfang der Distribution gehören, kann unterschiedlich sein: Im Idealfall existiert eine Paketquelle der Programmentwickler, die im Paketmanager eingebunden werden und über diesen dann installiert werden kann. Daneben gibt es für eine Reihe von Programmen Pakete, die auf die Distribution abgestimmt zum Download verfügbar sind. Im ungünstigsten Fall muss die Software als Quellcode bezogen werden und für das jeweilige System kompiliert werden. Anwendungen, die vom Anbieter nur für macOS oder Windows auf den Markt gebracht wurden, kann man i. d. R. unter Linux mittels API-Implementierungen wie Wine, Cedega oder Darling bzw. GNUstep verwenden. In anderen Fällen muss man zu alternativen Anwendungen greifen, die für Linux verfügbar sind.

Die beiden weit verbreiteten Desktop-Umgebungen *Gnome*<sup>8</sup> und *KDE*<sup>9</sup> haben unterschiedliche Bedienungskonzepte, weshalb viele Distributoren Standards und Richtlinien veröffentlichen, um sowohl Entwicklern als auch Nutzern den Umgang mit verschiedenen Desktop-Umgebungen nahezubringen und ihn zu vereinheitlichen.

Bekannt geworden sind größere Migrationen von Unternehmen oder Institutionen, die mehrere hundert oder tausend Rechner auf Linux-Desktops umgestellt haben, wie die Stadt München im Rahmen des *LiMux*<sup>10</sup>-Projekts oder die Umstellung von 20.000 Desktops bei Peugeot Citroën. Durch die Auslieferung vorinstallierter Systeme durch einige Fachhändler sowie die wachsende Beliebtheit einiger Distributionen wie Ubuntu wuchs die Linux-Verwendung auf Desktoprechnern von Anfang 2007 bis Mitte 2008 um fast 30 Prozent. In Großbritannien lag der Marktanteil 2008 bei etwa 2,8 Prozent. Weltweit wurde im April 2009 im Market-Share-Report von Net Applications erstmals ein Marktanteil von einem Prozent<sup>[18]</sup> ermittelt. Nachdem er 2010 gemäß NetMarketShare wieder auf 0,9 %<sup>[19]</sup> gefallen war, stieg der Marktanteil bis Dezember 2011 auf 1,41 %. Ende 2016 lag der Marktanteilanteil bei 2,2 %

---

<sup>8</sup> Siehe Kapitel Grafische Benutzeroberfläche

<sup>9</sup> Siehe Kapitel Grafische Benutzeroberfläche

<sup>10</sup> LiMux – Die IT-Evolution ist ein Projekt der Stadtverwaltung München, durch welches freie Software auf den derzeit rund 15.000 Arbeitsplatzrechnern der städtischen Mitarbeiter installiert wurde. Das Kofferwort LiMux setzt sich aus Linux und München zusammen. *Ergänzung:* Das Projekt wird mittlerweile als gescheitert angesehen.

---

## SERVER

Aufgrund der Kompatibilität von Linux mit anderen unixoiden Systemen hat sich Linux auf dem Servermarkt besonders schnell etabliert. Da für Linux schon früh zahlreiche häufig verwendete und benötigte Serversoftware wie Webserver, Datenbankserver und Groupware kostenlos und weitgehend uneingeschränkt zur Verfügung stand, wuchs dort der Marktanteil stetig.

Da Linux als stabil und einfach zu warten gilt, erfüllt es auch die besonderen Bedingungen, die an ein Server-Betriebssystem gestellt werden. Der modulare Aufbau des Linux-Systems ermöglicht zusätzlich das Betreiben kompakter, dedizierter Server. Außerdem hat die Portierung von Linux auf verschiedenste Hardwarekomponenten dazu geführt, dass Linux alle bekannten Serverarchitekturen unterstützt.

Eingesetzt wird es dabei für praktisch alle Aufgaben. Eines der bekanntesten Beispiele ist die Linux-Server-Konfiguration LAMP, bei der Linux mit Apache, MySQL und PHP/Perl (manchmal auch Python) kombiniert wird. Auch proprietäre Geschäftssoftware wie SAP R/3 ist mittlerweile auf verschiedenen Distributionen verfügbar und hat eine Installationszahl von über 1.000 Systemen erreicht. Das Linux Terminal Server Project ermöglicht es, sämtliche Software außer dem BIOS der Clients zentral zu verwalten.

Da Linux auf einer Vielzahl von verschiedenen Hardwaretypen betrieben werden kann, ist auch die für Linux-Server genutzte Hardware ähnlich umfangreich. Auch moderne Hardware wie die von IBMs eServer p5 wird unterstützt und ermöglicht dort das parallele Ausführen von bis zu 254 Linux-Systemen (Modell p595). Auf IBM-Großrechnern der aktuellen System-z-Linie läuft Linux wahlweise nativ, mittels PR/SM in bis zu 30 LPARs oder in jeder davon unter z/VM in potenziell unbegrenzt vielen, real einigen zehntausend virtuellen Maschinen.

Im Januar 2017 wurden mindestens 34 % aller Websites mittels eines Linux-Servers verfügbar gemacht. Da nicht alle Linux-Server sich auch als solche zu erkennen geben, könnte der tatsächliche Anteil um bis zu 31 Prozentpunkte höher liegen. Damit ist ein tatsächlicher Marktanteil von bis zu etwa 65 % nicht auszuschließen. Der Marktanteil von verkauften Linux-Server-Systemen lag im zweiten Quartal 2013 bei 23,2 %. Da bei Servern nicht selten von einem Kunden selbst ein anderes Betriebssystem installiert wird, gibt diese Zahl nur bedingt Auskunft über die effektive Verwendung von Linux auf Server-Systemen.

---

## SMARTPHONES UND TABLETS

Für Smartphones und Tablets gibt es speziell optimierte Linux-Distributionen. Sie bieten neben den Telefonie- und SMS-Funktionen diverse PIM-, Navigations- und Multimedia-Funktionen. Die Bedienung erfolgt typischerweise über Multi-Touch oder mit einem Stift. Linux-basierte Smartphonesysteme werden meist von einem Firmenkonsortium oder einer einzelnen Firma entwickelt und unterscheiden sich teilweise sehr stark von den sonst klassischen Desktop-, Embedded- und Server-Distributionen. Anders als im Embedded-Bereich sind Linux-basierte Smartphonesysteme aber nicht auf ein bestimmtes Gerät beschränkt, vielmehr dienen sie als Betriebssystem für Geräte ganz unterschiedlicher Modellreihen und werden oft herstellerübergreifend eingesetzt.

Die Architektur dieser Smartphone- und Tablet-Distributionen hat neben dem Linux-Kernel teilweise wenig mit den klassischen Distributionen zu tun. So wird von Android nur ein Teil der sonst üblichen GNU-Software-Umgebung genutzt. Die meist auf Linux genutzten UNIX-artigen Dienste und Tools werden teilweise durch eine Java-Laufzeitumgebung ersetzt. Dadurch entstehen neue Programmierschnittstellen, die sich auf beliebigen anderen Plattformen emulieren bzw. umsetzen lassen. Trotzdem wird Android als Linux-Distribution angesehen, die viele Eigenschaften mitbringt, die es mit zahlreichen Embedded-Linux-Distributionen teilt. Andere Smartphone-Distributionen, wie etwa Firefox OS, Ubuntu for phones, Maemo, Tizen, Mer, Sailfish OS[29] und MeeGo nutzen größere Teile der klassischen GNU-Software-Umgebung, so dass diese Distributionen teilweise einfacher mit klassischen Linux-Anwendungen ergänzt werden können und somit eher Linux-Distributionen im klassischen Sinne entsprechen.



Das von HP Palm entwickelte WebOS setzt ebenfalls auf dem Linux-Kernel auf, das Userland jedoch besteht aus einer proprietären Entwicklung unter anderer Lizenz. Auch das ehemals von Samsung entwickelte Bada war neben einem RTOS-Kernel auch auf einem Linux-Kernel nutzbar,[30] was aber von Samsung nie in dieser Kombination verkauft wurde.

Linux-Systeme haben seit Ende 2010 die Marktführerschaft auf dem schnell wachsenden Smartphone-Markt übernommen. Sie weisen in Deutschland seit Februar 2013 durchgehend einen Marktanteil von über 70 % auf mit einem bisherigen Maximum von über 82 % im Juli 2014 (Anteile Linux-basierter Alternativen zu Android wurden in der Statistik nicht explizit angegeben). Vorwiegend Android-Geräte haben Apple iOS, Windows Phone und Symbian OS erfolgreich zurückgedrängt.

## SYSTEMSTART

Der Bootvorgang selbst ist historisch gewachsen und wurde wegen der damaligen Platzprobleme (wenig Speicherplatz) auf viele Einzelschritte aufgeteilt, die Schritt-für-Schritt nachgeladen werden. Auf den meisten Rechnern, die vor 2012 verkauft wurden, kommt noch der BIOS-Modus zum Einsatz. Neuere Computer, insbesondere solche mit vorinstalliertem Windows 8, booten im EFI-Modus. Sofern EFI aktiv eingesetzt wird, kommt "intelligenter" Programmcode viel früher im Bootprozess zum Einsatz. EFI kann man dabei als eigenständiges, zwischengeschaltetes Betriebssystem verstehen, das umfangreiche Aufgaben übernehmen kann und weitere Komponenten in einem eigenen Dateisystem innerhalb des Ordners /EFI vorhält. Auch auf UEFI-Systemen ist weiterhin ein BIOS im Einsatz. Mehr Details sind der Artikelserie EFI Bootmanagement zu entnehmen.<sup>11</sup>

---

## ABGRENZUNG BIOS/ UEFI

Das alte BIOS wird nicht komplett durch UEFI ersetzt. Dinge wie der POST (Power On Self Test) oder die grundlegende Konfiguration über das BIOS-Setup werden weiterhin über das BIOS geregelt. Im weiteren Verlauf des Artikels wird der Bootvorgang nach dem BIOS- und/oder EFI-Modus unterschieden.

## GRAFISCHE BENUTZEROBERFLÄCHE<sup>12</sup>

**Desktop, Windowmanager** - wo ist da der Unterschied? Per Definition fallen unter die Bezeichnung "Desktop-Umgebung" Software-Pakete wie KDE, Gnome und Xfce. Diese Umgebungen bieten neben Fensterverwaltung, Desktop und Dateimanager auch eine Menge eigener Tools und Programme für den täglichen Gebrauch, um den Anwender wunschlos glücklich zu machen. Ihre spezielle Note erhalten Desktop-Umgebungen wie Gnome und KDE durch individuelle Bibliotheken und Toolkits, auf dem die mitgelieferten Programme basieren. Ein Toolkit stellt Programmierern eine Reihe von Werkzeugen und grafischen Elementen zur Verfügung, um möglichst schnell einheitliche Programmoberflächen zu bauen. Gnome basiert etwa auf dem GTK+-Toolkit, während die KDE-Entwickler bei ihrem Desktop auf Qt setzen. Zu den drei ausgewachsenen Komplett-Desktops gesellen sich viele Windowmanager wie Fluxbox, Enlightenment, Fvwm und weitere.

Ein Windowmanager ist die zentrale Komponente einer grafischen Benutzeroberfläche und verwaltet die dargestellten Fenster. Allerdings bringen Windowmanager nicht unbedingt auch einen Dateimanager oder eine Desktop-Verwaltung mit und sind deshalb schlank bis spartanisch. Dabei bleibt es dem Anwender überlassen, die Oberfläche mit zusätzlichen Tools und Programmen zu erweitern. Gerade das macht schlanke Windowmanager für Fortgeschrittene reizvoll, die einen kompromisslos schnellen Desktop wünschen. Aber auch auf älterer, ressourcen-schwacher Hardware sind die meist gewöhnungsbedürftigen Windowmanager eine echte Alternative.

---

<sup>11</sup> Vgl. <https://wiki.ubuntuusers.de/Bootvorgang/>, aufgerufen am 25.12.2017

<sup>12</sup> Vgl. <https://www.freenet.de/digitalewelt/windowssoftware/desktopvielfalt-unter-linux-372846-4734250.html>, aufgerufen am 25.12.2017

---

## DIE BASIS: X-WINDOW-SYSTEM

Die Vielfalt des Linux-Desktops beruht auf der Architektur des darunter liegenden Gerüsts, dem X-Window-System. Dessen Macher entschlossen sich 1984 dazu, einen möglichst offenen Standardbaukasten zu schaffen, der nur als Server agiert. Die grafischen Anwendungen verhalten sich dazu wie Clients. Definiert ist nur das verwendete Protokoll X11, nicht aber das Aussehen der Clients. Das führte zu einer Vielfalt unterschiedlicher Desktops, die sich in Aussehen und Funktionalität stark unterscheiden. Gut für den Linux-Anwender: Sie können mehrere Desktop-Varianten und Windowmanager parallel installieren und diese abwechselnd nutzen. Sie können auch Gnome-Anwendungen unter KDE starten und umgekehrt, Sie brauchen nur die passenden Bibliotheken. Um zwischen den installierten Desktops und Windowmanagern umzuschalten, sollten Sie einen Display-Manager wie Gdm oder Kdm verwenden, die bei den meisten Distributionen vorinstalliert sind. Damit können Sie am Log-in-Bildschirm auswählen, welche Oberfläche Sie für Ihre nächste Sitzung starten möchten.

---

## KDE: DER KONSISTENTE

1996 waren Desktops für Linux noch Mangelware. So kündigte der damals 24-jährige Student Matthias Ettrich im Usenet an, er wolle eine konsistente, nette und freie Desktop-Umgebung entwickeln. Heraus kam das "Kool Desktop Environment", das sich stark am damals unter Unix verbreiteten CDE orientierte, aber seinen eigenen Windowmanager Kwin nutzt. In der frühen Linux-Szene hagelte es sofort Kritik, da KDE auf dem proprietären Toolkit Qt der norwegischen Firma Trolltech basierte. Die Lizenz sahen viele Enthusiasten mit Linux unvereinbar und ignorierten KDE zunächst. Die Streitigkeiten mit der Open-Source-Gemeinde wurden erst 2000 beigelegt, als Qt in einer freien Linux-Version erschien. Der Durchbruch gelang KDE mit dem Browser Konqueror, der den instabilen Netscape Navigator ersetzte und noch vor Mozilla Marktreife erlangte.

KDE ist der Desktop, mit dem fast alle Umsteiger aus der Windows-Welt sofort zurechtkommen. Mit Compiz, Beryl und Compiz-Fusion kommen KDE-Anwender in den Genuss eines 3D-Desktops. Die meisten Distributionen liefern KDE mit, sogar Gnome-zentrierte Distributionen wie Debian und Fedora haben KDE im Gepäck. KDE ist allerdings nicht gerade sparsam in seinen Hardware-Anforderungen. Neben 250 MB Plattenplatz sollte der PC eine CPU mit Taktfrequenz im Gigahertz-Bereich sowie 512 MB RAM besitzen, wobei sich KDE zur Not auch mit weniger zufrieden gibt. Im Januar 2008 erschien KDE 4, das wegen grundlegender Änderungen nicht mehr zur 3.5-Serie kompatibel ist. Es basiert auf einer neuen Version von Qt und bringt eine vollkommen neue Optik sowie neue Anwendungen wie den Dateimanager Dolphin und den Betrachter Okular mit. Tipp: Ein standardmäßig mit Gnome ausgestattetes Ubuntu rüsten Sie im Nu mit KDE nach, indem Sie auf der Kommandozeile das Meta-Paket "kubuntu-desktop" nachinstallieren: "sudo apt-get install kubuntu-desktop".

---

## GNOME: DER DESKTOP-RIESE

Das GNU Network Object Model Environment - kurz Gnome - entstand 1997 als Antwort auf das damals noch proprietäre KDE und basiert auf dem Gimp-Toolkit GTK+. Gnome ist in der aktuellen Version 2.2 dabei schon lange kein Gnom mehr, sondern eigentlich ein speicherhungriges Monster. Die Oberfläche ist trotzdem einfach gehalten, was dem Desktop nicht nur von Linus Torvalds den Vorwurf einbrachte, seine Benutzer für dumm zu verkaufen. Der Vorteil der langsamen Entwicklung ist, dass Desktop und Anwendungen üblicherweise sehr ausgereift und stabil sind. Die Windowmanager Compiz, Beryl und Compiz-Fusion bringen dem Desktop sehenswerte 3D-Akrobatik bei.

Gnome läuft zwar auch auf schwächeren Rechnern, macht aber erst mit Hardware-Beschleunigung und der 3D-Erweiterung Compiz-Fusion richtig Spaß. Gnome ist inzwischen die wohl am häufigsten eingesetzte Desktop-Umgebung unter Linux. Der schlichte Desktop ist unter Ubuntu Standard und wird auch von Red Hat favorisiert. Mit IBM, HP und Sun beteiligen sich zudem potente Partner an der Gnome-Foundation. Bei den Hardware-Anforderungen liegt Gnome einen Tick unter denen von KDE und gibt sich mit einer schwächeren CPU unter 1 Gigahertz zufrieden, verlangt aber ebenfalls nach fast 512 MB RAM. Der Speicherbedarf ist hoch, da jedes noch so kleine Applet in einem eigenen Prozess läuft. Tipp: Möchten Sie mehr Einstellungsmöglichkeiten unter Gnome

nutzen, installieren Sie das Paket "gconf-editor" über den Paketmanager Ihrer Distribution oder von [www.gnome.org/projects/gconf](http://www.gnome.org/projects/gconf) (GConf-1.2.1.tar.bz2) nach. Damit passen Sie viele versteckte Eigenschaften selbst an.

---

## CINNAMON

Cinnamon (englisch für „Zimt“) ist eine freie Desktop-Umgebung für Linux. Das Projekt startete als Fork der Gnome-Shell, also als Grafische Shell, um unter Linux Mint eine Benutzeroberfläche zu schaffen, die moderne Konzepte von Gnome 3 mit der traditionellen Bedienung von Gnome 2 verbindet. Inzwischen ist Cinnamon auch für andere Linux-Distributionen verfügbar.

Mit Veröffentlichung der Desktop-Umgebung Gnome 3 wurde ein neues Design- und Bedienkonzept eingeführt, welches nicht zu den Vorstellungen der Linux-Mint-Entwickler passte. Daher wurde der Gnome-2-Fork MATE unter dem Dach von Linux Mint entwickelt. Um den Benutzern dennoch die Möglichkeit zu geben, die Distribution mit modernen Gnome-3-Features zu benutzen, stand auch diese Oberfläche zur Verfügung, wurde aber durch die Mint Gnome Shell Extensions (MGSE) äußerlich angepasst. Die Anpassungsfähigkeit der Gnome Shell durch Erweiterungen ist jedoch begrenzt, daher entschlossen sich die Entwickler schließlich, einen eigenen Gnome-3-Fork für ihre Bedürfnisse zu entwickeln, den sie Cinnamon nannten, nach dem englischen Wort für Zimt. Die auffälligste Änderung ist die Wiedereinführung der Taskleiste, deren Fehlen in Gnome 3 oft kritisiert wurde.

Ab Version 1.2 verwendet Cinnamon den Window Manager Muffin, eine Abspaltung von Mutter. Auch der von Gnome bekannte Dateimanager Nautilus wurde geforkt und kommt ab Cinnamon 1.6 unter dem Namen Nemo zum Einsatz. Ebenfalls neu ab dieser Version ist der Darstellungsmodus Cinnamon 2D, mit dem die Desktopumgebung auch auf Systemen ohne 3D-Beschleunigung funktioniert.

---

## MATE DESKTOP ENVIRONMENT

MATE ist eine freie Desktop-Umgebung für unixoide Systeme wie z. B. Linux. MATE entstand 2011 als Abspaltung der Desktop-Umgebung Gnome 2. Der Name leitet sich von der Mate-Pflanze ab[2] (spanisch yerba mate).

---

## XFCE: DER TUGENDHAFTE

Klein, charmant, komfortabel und zufrieden mit der ältesten Hardware: Xfce zeigt, dass eine umfassende Desktop-Lösung keine Gigahertz-CPU's braucht und 128 MB eine Menge Speicher sein können. Xfce basiert auf dem Gnome-Toolkit GTK+ 2.x, bringt aber seinen eigenen Windowmanager Xfwm mit. Den ergänzt es um zahlreiche Anwendungen, etwa den Dateimanager Thunar, den Editor Mousepad und komfortable Menüs für die Konfiguration. Seine Schlankheit erreicht Xfce dadurch, dass der Desktop außer dem Toolkit keine Gnome-Bibliotheken braucht und dass es bisher bei jedem großen Versionssprung von Grund auf neu geschrieben wurde. Außerdem halten sich die Programmierer strikt an die Standards, die das Freedesktop-Gremium empfiehlt.

Zwar ist Xfce schon eine Weile in den Fedora- und Debian-Repositories. Zu neuem Ruhm gelangte der Desktop aber mit Xubuntu, das als erste prominente Distribution Xfce ins Scheinwerferlicht rückte. Obwohl die Programmierer dem Desktop immer mehr eigene Anwendungen und optische Effekte spendieren, bleibt er bescheiden: Eine CPU mit 300 MHz sowie 128 MB RAM sind genug. Tipp: Der Windowmanager von Xfce unterstützt seit der Version 4.4.2 als einziger Desktop den eingebauten Compositor von X, der Transparenz und Schattenwurf bietet. Um ihn einzuschalten, wählen Sie im Xfce-Menü den Punkt "Einstellungen, Fenstermanager-Tweaks, Compositor". Übrigens lässt sich ein Ubuntu-System über die Kommandozeile ganz einfach mit dem Befehl "apt-get install xubuntu-desktop" um Xfce ergänzen.

---

## HILFREICHE LINKS

Wiki Ubuntuusers (<https://wiki.ubuntuusers.de/>)

Linux Wiki ([www.linuxwiki.de/](http://www.linuxwiki.de/))

Website des Linux Mints Projekts (<https://linuxmint.com/>)

Linux Mint Foren (<https://forums.linuxmint.com/>)

## DATEIVERWALTUNG

Dateisysteme sind die Schnittstellen zwischen dem Betriebssystem und den Partitionen auf Datenträgern. Sie organisieren die geordnete Ablage von Daten. Als Benutzer bekommt man von der Arbeit eines Dateisystems nicht viel mit. Man sieht nur, dass Dateien oder Verzeichnisse vorhanden sind, nicht aber wo und wie diese Daten auf dem Datenträger organisiert, gespeichert werden. Diese Arbeit übernimmt das Dateisystem. Es richtet eine Art Inhaltstabelle ein, in der alle Dateien (auch Verzeichnisse) und ihre Speicheradressen verwaltet werden. Neben einer solchen grundlegenden Datenorganisation stellen die verschiedenen Dateisysteme noch unterschiedliche zusätzliche Möglichkeiten zur Verfügung. Dieser Artikel verdeutlicht einige der Unterschiede zwischen den verschiedenen Dateisystemen.<sup>13</sup>

Linux nutzte bis ca. 1992 das Dateisystem *Minix*<sup>14</sup>, was aber aus unterschiedlichen Gründen (max. Partitionsgröße 64 MB, Dateinamen durften nur 14 Zeichen lang sein) nicht mehr den Ansprüchen genügte. Als Alternative bot sich das ext-Dateisystem an, welches (fast) allen Anforderungen gerecht wurde.

## DATENTRÄGERVERWALTUNG<sup>15</sup>

Umsteiger von Windows auf Linux fragen sich oft, wo die gewohnten Laufwerksbuchstaben zu finden sind. Die Antwort ist einfach: es gibt sie nicht. Datenträger und Dateisysteme werden unter Linux ganz anders und wesentlich flexibler gehandhabt.

Datenträger lassen sich auf zweierlei Arten beschreiben:

- Sie befinden sich physikalisch an einem bestimmten Ort (Partition einer Festplatte, CD-ROM, Netzwerkfreigabe, MP3-Player,...)
- Sie werden im System an einer bestimmten Stelle verwendet (C:, D:, E:, ... oder /home, /home/benutzer/mp3, /media/cdrom, /media/camera, ...)

Die ersten dieser Beispiele sind die von Windows bekannten Buchstaben, die letzteren sind schlicht Verzeichnisse in einem Linux-System. Beliebige Verzeichnisse? Im Prinzip ja. Unter Linux lässt sich der Ort, an dem man auf einen Datenträger zugreifen kann, beliebig festlegen. An der Stelle des Verzeichnisses /home/benutzer/mp3 könnte sich technisch gesehen eine Festplattenpartition befinden, eine CD-ROM, ein mobiles Gerät, eine USB-Platte - ganz wie man es gerade benötigt.

Wozu ist das nun gut? Ein Beispiel: Bei der Installation des Betriebssystems wurde einfach alles auf eine einzige Partition gelegt, das System wie auch die Dateien der einzelnen Benutzer. Was geschieht, wenn man jetzt diese Benutzerdaten auf eine andere Partition verlagern will? Unter Windows sind die Daten dann unter einem anderen Laufwerksbuchstaben erreichbar - statt C:\Dokumente und Einstellungen heißt es D:\Dokumente und Einstellungen. Und unter Linux? /home bleibt /home, ob darunter nun die einzige Partition der einzigen Festplatte liegt, die dritte Partition auf dem externen *FireWirelaufwerk*<sup>16</sup> oder eine Netzwerkfreigabe auf irgendeinem Server.

<sup>13</sup> Vgl. <https://wiki.ubuntuusers.de/Dateisystem/>, aufgerufen am 25.12.2017

<sup>14</sup> Das Minix file system ist das Dateisystem des ehemaligen Lehr-Betriebssystems Minix.

<sup>15</sup> Vgl. <https://wiki.ubuntuusers.de/Dateisystem/>, aufgerufen am 25.12.2017

<sup>16</sup> FireWire, i.LINK oder 1394 ist ein Bus für serielle Datenübertragung.

Die Geräte sind unter Linux nach einem einfachen Schema benannt: Zuerst steht immer das Verzeichnis `/dev/`. In diesem Verzeichnis finden sich alle "Gerätedateien"; über Gerätedateien lässt sich die Hardware ansprechen (Anmerkung: Unter Linux ist alles eine Datei, was Daten annehmen und/oder liefern kann - also buchstäblich alles. Das ist eins der ebenso simplen wie genialen Grundprinzipien von unixartigen Systemen).

**An nächster Stelle kommt die Art des Geräts:**

- SATA-Festplatten (neuere interne Festplatten) und IDE-Festplatten (ältere interne Festplatten) beginnen mit den Buchstaben `sd`
- CD/DVD-Laufwerke (*IDE*<sup>17</sup>, *S-ATA*<sup>18</sup>) beginnen mit `sr`
- SCSI-Festplatten und alle externen Speichermedien, die über USB oder Firewire angeschlossen sind, beginnen ebenfalls mit `sd`
- externe oder SCSI-CD/DVD-Laufwerke beginnen mit `scd`

**Dann kommt der Anschluss an die Reihe - es gibt ja mehrere S-ATA-, IDE/ATAPI- oder SCSI-Anschlüsse:**

- bei SATA- und SCSI-Festplatten werden die Buchstaben der Reihe nach verwendet (`/dev/sda`, `/dev/sdb`, `/dev/sdc` usw.)
- bei IDE-Geräten (interne Festplatten und CD-ROM-Laufwerke) ist wichtig, mit welchem IDE-Anschluss das Gerät verbunden ist. Jeder Anschluss kann zwei Geräte aufnehmen (sogenannte "Master" und "Slave"; bei S-ATA nicht relevant):
- SCSI- oder externe CD-ROMS werden mit Zahlen bei 0 beginnend numeriert (`/dev/scd0`, `/dev/scd1` usw.)

**Festplatten sind darüber hinaus in Partitionen unterteilt. Es gibt zwei Arten von Partitionen:**

- die "klassischen" primären Partitionen werden von 1 bis 4 numeriert (`/dev/sda1` bis `/dev/sda4`)
- eine der primären Partitionen kann als erweiterte Partition weitere "Partitionen", die sogenannten logischen Laufwerke enthalten. Deren Benennung beginnt in jedem Fall bei der Ziffer 5 (`/dev/sda5`, `/dev/sda12` usw.)
- RAID-Geräte beginnen mit `md` und werden dann mit 0 beginnend hochgezählt (`/dev/md0`, `/dev/md1` usw.)
- Logische Volumes, wie sie von LVM oder EVMS erzeugt werden, finden sich an gesonderter Stelle. Hierzu empfiehlt sich die Lektüre der jeweiligen Anleitung.

Wie kann man sich das auf dem Terminal ausgeben?

```
| cat /proc/partitions
```

## DAS DATEISYSTEM<sup>19</sup>

### BTRFS-DATEISYSTEM

Dieses Dateisystem (Art der Formatierung eines Datenträgers) beinhaltet ein Copy-On-Write System sowie eine neuartige Daten- / Metaverwaltung (das sogenannte Journaling), das von der Firma Oracle Corporation seit dem

<sup>17</sup> Die Bezeichnung Integrated Drive Electronics (kurz IDE) stammt aus der Zeit vor der Standardisierung und wird üblicherweise synonym zu ATA bzw. PATA benutzt

<sup>18</sup> Serial ATA (Serial AT Attachment) – üblich sind auch die Schreibweisen als Abkürzung, SATA und S-ATA – ist eine Computer-Schnittstelle für den Datenaustausch mit Festplatten und anderen Speichergeräten. ATA steht hierbei für das Übertragungsprotokoll „AT Attachment“ – im Unterschied dazu werden bei SATA die Daten jedoch seriell (englisch serial) übertragen.

<sup>19</sup> Vgl. <https://wiki.ubuntuusers.de/Dateisystem/>, aufgerufen am 25.12.2017

Jahr 2007 für das Betriebssystem Linux entwickelt wurde und dann mit dem Kernel 2.6.29 allgemein eingeführt wurde.

Es soll in Zukunft die bislang im Linux-Umfeld vorherrschenden Dateisysteme ext3 bzw. ext4 mit seinen Beschränkungen (Dateigröße und Gesamtdateisystemgröße) ersetzen.

---

## EXT-DATEISYSTEM

Im Gegensatz zu den unter Windows verwendeten Dateisystemen FAT32 und NTFS, die auf einer Tabellenstruktur basieren, verwenden ext-Dateisysteme eine Baumstruktur zum Speichern des Laufwerksinhaltes. Die Knoten dieser Baumstruktur bestehen aus Inodes (Kurzwort aus "index node"). Diese Inodes speichern entweder direkte Verweise auf Festplattenblöcke mit kleinen Dateien oder einen Verweis auf einen weiteren Block, in dem die Datenblöcke größerer Dateien gespeichert sind. Ein Inode hat eine Mindestgröße von 128 Byte.

Informationen über das Dateisystem werden im ersten Block, im Superblock des Dateisystems, gespeichert. Hier finden sich alle Optionen des Dateisystems, die sich auch mit dem Befehl `tune2fs` manipulieren lassen. Außerdem findet sich dort ein Verweis auf die Inode-Struktur des zu Grunde liegenden Dateisystems.

---

## WEITERE

Untenstehend eine Tabelle mit allen bekannte und verwendeten Dateisystemen. Links führen zu ausführlichen Artikel aus dem Web.

Dateisystem	Veröffentlicht	Entwickler	Entwickelt für Betriebssystem	Links
<a href="#">btrfs</a>	2007	Oracle	Linux	<a href="#">btrfs</a>
ext2	1993	Rémy Card u.a.	Linux	<a href="#">ext2</a>
ext3	2001	Stephen Tweedie u.a.	Linux	<a href="#">ext3</a>
ext4	2008	verschiedene	Linux	<a href="#">ext4</a>
JFS	1990	IBM	AIX	<a href="#">JFS</a>
ReiserFS	2001	Namesys	Linux	<a href="#">ReiserFS</a>
Reiser4	2004	Namesys	Linux	<a href="#">ReiserFS</a>
XFS	1994	SGI	Unix	<a href="#">XFS</a>
FAT	1980	Microsoft	MS-DOS	<a href="#">FAT</a>
<a href="#">exFAT</a>	2006	Microsoft	Windows	<a href="#">exFAT</a>
NTFS	1993	Microsoft	Windows	<a href="#">NTFS</a>
<a href="#">ZFS</a>	2006	Sun	Solaris	<a href="#">ZFS</a>

## DIE VERZEICHNISSTRUKTUR<sup>20</sup>

---

<sup>20</sup> Vgl. <https://wiki.ubuntuusers.de/Verzeichnisstruktur/>, aufgerufen am 25.12.2017

In verschiedenen Systemverzeichnissen liegen alle elementaren Programme und Konfigurationsdateien. In der Regel können in diesen Verzeichnissen (Ordern) nur mit Root-Rechten Veränderungen vorgenommen werden. Obwohl dies gerade Ein- und Umsteigern gerne Probleme bereitet, ist dieser Zwang ein essentieller Baustein der Systemsicherheit.

Lange Zeit brachte jede Linux-Distribution ihre eigenen Vorstellungen vom Aufbau des Dateisystems mit. Um eine Vereinheitlichung und den Datenaustausch zwischen den Distributionen zu erleichtern, hat man ein Gremium aus den wichtigsten Distributoren gebildet. Dieses Gremium erarbeitete den "Filesystem Hierarchy Standard" (FHS). Auch die Verzeichnisstruktur von Ubuntu richtet sich nach diesem Standard.

Die hier gezeigte Verzeichnisstruktur berücksichtigt eine Installation mit gesonderter Bootpartition nicht. Die Auflistung erfolgt in alphabetischer Reihenfolge. Noch mehr Informationen - auch über viele Unterverzeichnisse - kann man der Manpage von hier entnehmen. Das Wort "hier" steht in diesem Fall für "Description of the file system hierarchy", übersetzt, "Beschreibung der Hierarchie des Dateisystems".

---

## / (ROOT-VERZEICHNIS)

Erste Verzeichnisebene; entspricht in der Regel der Bootpartition (Ausnahme: Eigene Bootpartition); enthält neben den im Folgenden genannten Unterverzeichnissen auch symbolischen Verknüpfungen für `initrd.img` und `vmlinuz` (siehe `/boot`)

---

## /BIN

Von: binaries (Programme); muss bei Systemstart vorhanden sein; enthält für Linux unverzichtbare Programme; diese Programme können im Gegensatz zu `/sbin` von allen Benutzern ausgeführt werden; `/bin` darf keine Unterverzeichnisse enthalten.

---

## /BOOT

Muss bei Systemstart vorhanden sein; Enthält zum Booten benötigte Dateien. Beispiele: Kernel "`vmlinuz-versionnummer-generic`" (für Desktopsysteme), "`vmlinuz-versionnummer-server`" (für Serversysteme), "`vmlinuz-versionnummer-virtual`" (für virtuelle Maschinen), initiale Ramdisk "`initrd.img-versionnummer-generic`"/"`-server`"/"`-virtual`" und das Programm für den Memorytest `memtest86.bin`

### Unterverzeichnis(se):

- `/boot/grub`: Enthält Dateien der Bootloader GRUB 2 und GRUB
- `/boot/efi`: Enthält EFI-Programme - nur bei eingesetztem und aktiv genutztem EFI

---

## /DEV

Von: devices (Geräte); muss bei Systemstart vorhanden sein; enthält alle Gerätedateien, über die die Hardware im Betrieb angesprochen wird; Gerätedateien für Hot-Plug-Systeme (z.B. USB, IEEE1394) werden von `udev` erstellt. Besondere Verzeichnisse:

- `/dev/null`: Kann man als Papierkorb sehen; alle Dateien, die dort hin verschoben werden, werden vernichtet.
- `/dev/random`: Wenn man dieses Verzeichnis ausgibt mit „cat“ bekommt man Zufallsstrings.

---

## /ETC

Von: et cetera ("alles übrige"), später auch: editable text configuration (änderbare Text Konfiguration); muss bei Systemstart vorhanden sein; enthält Konfigurations- und Informationsdateien des Basissystems. Beispiele: `fstab`,

hosts, lsb-release, blkid.tab; hier liegende Konfigurationsdateien können durch gleichnamige Konfigurationsdateien im Homeverzeichnis überlagert werden. Beispiel: bash.bashrc

#### Unterverzeichnisse u.a.:

- **/etc/default:** Enthält u.a. die Konfigurationsdatei grub
- **/etc/grub.d:** Enthält Skripte für Grub\_2
- **/etc/opt:** Verzeichnisse und Konfigurationsdateien für Programme in /opt
- **/etc/X11:** Verzeichnisse und Konfigurationsdateien des XServer; Beispiel: xorg.conf
- **/etc/network:** Verzeichnisse und Konfigurationsdateien des Netzwerkes: Beispiel interfaces
- **/etc/init.d:** Enthält Start- und Stopskripte; Siehe Dienste
- **/etc/sgml:** Konfigurationsdateien für SGML (optional)
- **/etc/xml:** Konfigurationsdateien für XML (optional)

---

#### /HOME

Das Verzeichnis enthält pro Benutzer ein Unterverzeichnis; jedes Verzeichnis wird nach dem Anmeldenamen benannt. Das Homeverzeichnis ist der Ort, an dem Benutzer ihre Daten ablegen können und an dem Programme ihre benutzerspezifischen Einstellungen hinterlegen. Nur hier hat der einzelne Benutzer volle Schreib- und Leserechte. Und nur hier sollten Benutzer ihre Daten speichern. Wichtig ist hierbei, dass die Groß-/Kleinschreibung von Ordnern und Dateien unter Linux – im Gegensatz zu Windows – berücksichtigt wird. Beispiel.doc und beispiel.doc sind also zwei unterschiedliche Dateien.

Einstellungen werden üblicherweise in versteckten Dateien und Verzeichnissen gespeichert. Diese erkennt man daran, dass die Namen mit einem Punkt beginnen. .bashrc oder .gnome2 sind Beispiele für solche Konfigurationsdateien bzw. -verzeichnisse. Daneben können versteckte Dateien oder Verzeichnisse auch sehr sensible Daten wie persönliche Schlüssel oder die verschlüsselten eigenen Dateien enthalten. Im Umgang mit versteckten Dateien und Verzeichnissen sollte man deshalb die gleiche Sorgfalt walten lassen wie mit nicht-versteckten.

---

#### /LIB

Von: libraries (Bibliotheken); muss bei Systemstart vorhanden sein; enthält unverzichtbare Bibliotheken fürs Booten und die dynamisch gelinkten Programme des Basissystems; mindestens notwendige Dateien bzw. symbolische Verknüpfungen: libc.so.\* (C-Bibliotheken), ld\* (Bibliotheksdaten: Versionen, Zugriffspfade,...)

---

#### /LOST+FOUND

(verloren und gefunden); Dateien und Dateifragmente, die beim Versuch, ein defektes Dateisystem zu reparieren, übrig geblieben sind. Das Verzeichnis ist auf Partitionen mit reiserfs oder xfs nicht vorhanden. Siehe auch Datenrettung.

---

#### /MNT

Von: mount (eingehängt); normalerweise leer; kann für temporär eingehängte Partitionen verwendet werden. Für Datenträger, die hier eingehängt werden, wird im Gegensatz zu /media kein Link auf dem Desktop angelegt (gilt nur für GNOME).



---

## /OPT

Von: optional; ist für die manuelle Installation von Programmen gedacht, die ihre eigenen Bibliotheken mitbringen und nicht zur Distribution gehören; siehe /opt; Speicherort für selbst erstellte Programme siehe /usr/local in /usr

---

## /ROOT

Ist das Homeverzeichnis des Superusers (root). Der Grund, wieso sich das /root-Verzeichnis im Wurzelverzeichnis und nicht im Verzeichnis /home befindet, ist, dass das Homeverzeichnis von Root immer erreichbar sein muss, selbst wenn die Home-Partition aus irgendeinem Grund (Rettungs-Modus, Wartungsarbeiten) mal nicht eingehängt ist.

---

## /USR

Von: user (siehe: Herkunft); enthält die meisten Systemtools, Bibliotheken und installierten Programme; der Name ist historisch bedingt - früher, als es /home noch nicht gab, befanden sich hier auch die Benutzerverzeichnisse. **Unterverzeichnisse u.a.:**

- **/usr/bin** : Anwenderprogramme; Hier liegen die Desktopumgebungen und die dazu gehörigen Programme, aber auch im Nachhinein über die Paketverwaltung installierte Programme, wie Audacity. Hier liegen aber nicht die manuell installierten Programme (die in /opt installiert werden).
- **/usr/include** : Header-Dateien für C-Programme (nur für Programmierer interessant)
- **/usr/lib** : Bibliotheken
- **/usr/local** : Das Verzeichnis **/usr/local** enthält noch einmal die gleiche Verzeichnisstruktur wie **/usr** und ist für Programme gedacht, die man an der Paketverwaltung vorbei installieren möchte, z.B. selbst kompilierte Programme. Bei einem neu installierten Ubuntu enthält es höchstens leere Ordner.
- **/usr/sbin** : für Linux nicht unbedingt notwendige Systemprogramme; Beispiel: dmidecode
- **/usr/share** : im Gegensatz zu /var für statische, sich nicht ändernde Dateien; ist Architektur unabhängig, nicht aber vom Betriebssystem (z.B. können sich "Ubuntu 8.10 i386" und "Ubuntu 8.10 amd64" dieses Verzeichnis teilen, aber nicht "Ubuntu 8.10" und "Ubuntu 9.04")
- **/usr/share/applications** : Programmstarter, die für Anwendungsmenüs genutzt werden
- **/usr/share/man** : Manpages

---

## /VAR

Von **variable** (variabel); enthält nur Verzeichnisse; Dateien in den Verzeichnissen werden von den Programmen je nach Bedarf geändert (im Gegensatz zu [/etc](#)); Beispiele: Log-Dateien, Spielstände, Druckerwarteschlange

Unterverzeichnisse u.a.:

- **/var/log** : Alle Log-Dateien der Systemprogramme; Beispiele: **Xorg.0.log** (Log-Datei des XServer), **kern.log** (Logdatei des Kernels), **dmesg** (letzte Kernelmeldungen), **messages** (Systemmeldungen); Siehe auch Logdateien
- **/var/games** : Spielstände von Linuxspielen
- **/var/run** : Dateien, die den Zustand des Gesamtsystems speichern; Inhalt wird bei jedem Booten gelöscht und neu geschrieben (siehe auch /run)
- **/var/tmp** : Daten, die bei einem Reboot zwischengespeichert werden müssen (nicht verlorengehen sollen)
- **/var/www** : Im FHS nicht vorhanden, bei Ubuntu das "document root" des Webserver Apache

- **/var/spool** : Druckaufträge

## NÜTZLICHE TOOLS

## RECHTEVERWALTUNG<sup>21</sup>

Linux-Dateisysteme besitzen standardmäßig ein Rechtssystem, welches die Rechte pro Dateien und Verzeichnissen festlegt. Zum Ändern dieser Rechte können entweder die Terminal-Befehle `chmod`, `chown` sowie `chgrp` oder aber die Standard-Dateimanager Nautilus (Ubuntu/Ubuntu GNOME), Dolphin (Kubuntu), Caja (Ubuntu MATE), Thunar (Xubuntu) und PCManFM (Lubuntu) verwendet werden.

Prinzipiell können Benutzer immer nur die Dateirechte solcher Dateien ändern, deren Besitzer sie selbst sind. Sollen Rechte von Dateien anderer Benutzer oder von Systemdateien verändert werden, so sind dafür stets Root-Rechte erforderlich.

Standardmäßig existiert unter Linux immer ein Konto für den Benutzer `root` mit der User-ID 0. Dies ist ein Systemkonto mit vollem Zugriff auf das gesamte System und damit auch auf alle Dateien und Einstellungen aller Benutzer. Bei Linux Mint wird dem Benutzer `root` allerdings kein Passwort zugewiesen. Dadurch kann sich niemand unter dem Namen "`root`" anmelden.

## SUDO & ROOT – KONZEPT DER RECHTE

Der Befehl `sudo` kann Programmaufrufen vorangestellt werden. Er ermöglicht berechtigten Benutzern, das Programm im Namen und mit den Rechten eines anderen Benutzers auszuführen. Beispielsweise um Aufgaben auszuführen, die Administratoren vorbehalten sind:

- Programme installieren (`sudo apt-get install ...`)
- Systemkonfigurationen ändern (`sudo nano /etc/fstab`)

`Sudo` fragt vor der Ausführung des Programms unter einem anderem Namen nach dem Passwort des aufrufenden Benutzers. Damit wird überprüft, ob dieser den Befehl selbst eingegeben hat. Der Kreis der berechtigten Benutzer ist in der Datei `/etc/sudoers` festgelegt. Standardmäßig wird als Ziel-Benutzer `root` angenommen.

## BENUTZER UND GRUPPEN

Linux kennt als Mehrbenutzer-Betriebssystem – wie alle unixoiden Betriebssysteme – das Konzept verschiedener Benutzer. Diese haben nicht alle unbedingt dieselben Rechte und Privilegien. Neben den eigentlichen Benutzerkonten für reale Personen existieren auf dem System noch viele Systemdienste mit einem eigenen Benutzerkonto. Dadurch wird erreicht, dass eine mögliche Schwachstelle in einem Dienst nicht zu große Auswirkungen auf das System haben kann.

Im Gegensatz zu vielen anderen Linux-Distributionen wird unter Linux Mint dem Systemverwalterkonto "`root`" kein Passwort zugewiesen, d.h. es gibt kein "klassisches" Administrator-Konto. Dies ist Absicht.

Jeder Benutzer ist einer Hauptgruppe zugeordnet, kann daneben aber auch Mitglied weiterer Gruppen sein. Der Zugriff auf gewisse Hardware oder Dienste ist auf die Mitglieder einer bestimmten Gruppe beschränkt. So dürfen z.B. nur Benutzer, die zur Gruppe "`audio`" gehören, Klänge über die Soundkarte ausgeben. Möchte man nun einem Benutzer die Berechtigung für die Soundkarte geben, so erreicht man dies, indem man ihn in die Gruppe "`audio`" aufnimmt.

Die Hauptgruppe eines Benutzers spielt im Zusammenhang mit den Zugriffsberechtigungen auf Dateien und Verzeichnisse eine Rolle. Jede Datei ist immer Eigentum genau eines Benutzers. Daneben ist den Dateien aber auch eine Gruppe zugeordnet. Wenn ein Benutzer eine Datei erzeugt, dann wird seine Hauptgruppe als Gruppe bei der erzeugten Datei eingetragen.

---

<sup>2121</sup> Vgl. <https://wiki.ubuntuusers.de/Rechte/>, aufgerufen am 25.12.2017

DATEISYSTEMRECHTE<sup>22</sup>

UNIX-Systeme wie Linux verwalten ihre Dateien in einem virtuellen Dateisystem (VFS, Virtual File System). Dieses ordnet jeder Datei über eindeutig identifizierbare Inodes unter anderem folgende Eigenschaften zu:

- Dateityp (einfache Datei, Verzeichnis, Link, ...)
- Zugriffsrechte (Eigentümer-, Gruppen- und sonstige Rechte)
- Größe
- Zeitstempel
- Verweis auf Dateiinhalt

Jedes unter Linux gängige UNIX-Dateisystem (z.B. ext2/3/4, ReiserFS, xfs usw.) unterstützt diese Rechte. Gar nicht umgesetzt werden die Rechte hingegen auf VFAT-Dateisystemen; dort können Dateirechte lediglich beim Einhängen simuliert werden. Partitionen mit dem Windows-Dateisystem NTFS werden zwar in Linux standardmäßig ähnlich wie VFAT-Partitionen behandelt; mit den Mount-Optionen `permissions` und `acl` lässt sich aber auch auf NTFS-Partitionen eine echte Rechteverwaltung wie bei UNIX-Dateisystemen einrichten. Siehe hierzu Windows-Partitionen einbinden sowie NTFS-3G.

Im Terminal lassen sich die Rechte mit dem Befehl `ls -l` anzeigen. Im Folgenden sind als Beispiel die Dateirechte des Verzeichnisses `/var/mail/` dargestellt (Befehl: `ls -ldh /var/mail/`):

```
| drwxrwsr-x 2 root mail 4,0K Apr 23 2012 /var/mail/
```

Nach dem Dateityp kommen drei Zeichengruppen zu je drei Zeichen. Diese kennzeichnen die Zugriffsrechte für die Datei bzw. das Verzeichnis. Hat der Benutzer/Gruppe/andere ein Recht, so wird der Buchstabe dafür angezeigt; ansonsten wird ein - dafür angezeigt.

In obigen Beispiel erscheint nach dem Dateityp dann die Zeichenfolge `rw-rwsr-x`. Wenn man diese in drei Dreiergruppen aufteilt, erhält man diese Gruppen:

- `rw`: Rechte des Eigentümers
- `rw`: Rechte der Gruppe
- `r-x`: Recht von allen anderen (others)

Zeichen	Bedeutung	Beschreibung
<b>r</b>	Lesen (read)	Erlaubt lesenden Zugriff auf die Datei. Bei einem Verzeichnis können damit die Namen der enthaltenen Dateien und Ordner abgerufen werden (nicht jedoch deren weitere Daten wie z.B. Berechtigungen, Besitzer, Änderungszeitpunkt, Dateiinhalt etc.).
<b>w</b>	Schreiben (write)	Erlaubt schreibenden Zugriff auf eine Datei. Für ein Verzeichnis gesetzt, können Dateien oder Unterverzeichnisse angelegt oder gelöscht werden, sowie die Eigenschaften der enthaltenen Dateien/Verzeichnisse verändert werden.
<b>x</b>	Ausführen (execute)	Erlaubt das Ausführen einer Datei, wie das Starten eines Programms. Bei einem Verzeichnis ermöglicht dieses Recht, in diesen Ordner zu wechseln und weitere Attribute zu den enthaltenen Dateien abzurufen (sofern man die Dateinamen

<sup>22</sup> Vgl. <https://wiki.ubuntuusers.de/Rechte/>, aufgerufen am 25.12.2017

		kennt ist dies unabhängig vom Leserecht auf diesen Ordner). Statt x kann auch ein <a href="#">Sonderrecht</a> angeführt sein.
--	--	---

## DARSTELLUNGSARTEN

Neben der symbolischen Darstellung (z.B. `rw-rw-r-x`) gibt es auch noch eine oktale Darstellung (nach dem Oktalsystem). Die Grundrechte (Lesen, Schreiben, Ausführen) und Kombinationen daraus werden hierbei durch eine einzelne Ziffer repräsentiert und dem Eigentümer, der Gruppe und allen anderen zugeordnet. Je nach Anwendung wird dabei von unterschiedlichen Grundwerten ausgegangen und entweder Rechte gegeben oder entzogen. Bei `chmod` wird beispielsweise von der Grundeinstellung „keine Rechte“ (000) ausgegangen und Rechte gegeben, wohingegen bei `umask` von „alle Rechte vorhanden“ (777) ausgegangen und Rechte entzogen werden. Entsprechend sind die Werte je nach Anwendung anders.

Mögliche Werte für				
	chmod (octal)	umask (octal)	Symbolisch	Binäre Entsprechung
<b>Lesen, schreiben und ausführen</b>	7	0	<code>rx</code>	111
<b>Lesen und Schreiben</b>	6	1	<code>rw-</code>	110
<b>Lesen und Ausführen</b>	5	2	<code>r-x</code>	101
<b>Nur lesen</b>	4	3	<code>r--</code>	100
<b>Schreiben und Ausführen</b>	3	4	<code>-wx</code>	011
<b>Nur Schreiben</b>	2	5	<code>-w-</code>	010
<b>Nur Ausführen</b>	1	6	<code>--x</code>	001
<b>Keine Rechte</b>	0	7	<code>---</code>	000

Hier ein paar Beispiele:

- `rw-rw-rwx` entspricht 0777 (`chmod`) oder 0000 (`umask`): Jeder darf lesen, schreiben und ausführen.
- `rw-r-xr-x` entspricht 0755 (`chmod`) oder 0022 (`umask`): Jeder darf lesen und ausführen, aber nur der Dateibesitzer darf diese Datei (oder das Verzeichnis) auch verändern.

Bei folgenden Beispiele behandeln zusätzlich die [Sonderrechte](#). Diese Rechte können nicht per `umask` gesetzt werden:

- `rwsr-xr-x` entspricht 4755 (`chmod`): Jeder darf lesen und ausführen, der Eigentümer darf diese Datei oder das Verzeichnis verändern. Zusätzlich ist noch das Set-UID-Recht gesetzt.
- `rw-r--r-T` entspricht 1644 (`chmod`): Jeder darf lesen, der Eigentümer darf schreiben. Zusätzlich ist das Sticky-Bit gesetzt (aber nicht das Ausführrecht, daher ein großes T).

## SONDERRECHTE

Die oben gezeigten Dateirechte kann man als Basisrechte bezeichnen. Für besondere Anwendungen gibt es zusätzlich noch besondere Dateirechte. Der Einsatz dieser ist nur dann ratsam, wenn man genau weiß, was man tut, da dies unter Umständen zu Sicherheitsproblemen führen kann.

Sonderrechte			
Zeichen	Bedeutung	Beschreibung	
<b>s</b>	Set-UID-Recht (SUID-Bit)	Das Set-UID-Recht („Set User ID“ bzw. „Setze Benutzerkennung“) sorgt bei einer Datei mit Ausführungsrechten dafür, dass dieses Programm immer mit den Rechten des Dateibesitzers läuft. Bei Ordnern ist dieses Bit ohne Bedeutung.	
<b>s (S)</b>	Set-GID-Recht (SGID-Bit)	Das Set-GID-Recht („Set Group ID“ bzw. „Setze Gruppenkennung“) sorgt bei einer Datei mit Ausführungsrechten dafür, dass dieses Programm immer mit den Rechten der Dateigruppe läuft. Bei einem Ordner sorgt es dafür, dass die Gruppe an Unterordner und Dateien vererbt wird, die in diesem Ordner neu erstellt werden.	
<b>t (T)</b>	Sticky-Bit	Das Sticky-Bit („Klebrig“) hat auf modernen Systemen nur noch eine einzige Funktion: Wird es auf einen Ordner angewandt, so können darin erstellte Dateien oder Verzeichnisse nur vom Dateibesitzer gelöscht oder umbenannt werden. Verwendet wird dies z.B. für <b>/tmp</b> .	

Die Symbole für die Sonderrechte erscheinen an der dritten Stelle der Zugriffsrechte, die normalerweise dem Zeichen x (für executable) vorbehalten ist, und ersetzen ggf. dieses. Die Set-UID/GID-Rechte werden anstelle des x für den Besitzer bzw. die Gruppe angezeigt, das Sticky-Bit anstelle des x für andere. Wenn das entsprechende Ausführrecht gesetzt ist, wird ein Kleinbuchstabe verwendet, ansonsten ein Großbuchstabe.

## NÜTZLICHE TOOLS

## PAKETVERWALTUNG<sup>23</sup>

Eine Softwarepaket-Verwaltung (englisch package management software) ermöglicht die komfortable Verwaltung von Software, die in Programmpaketform vorliegt, auf einem Betriebssystem. Dazu gehören das Installieren, Aktualisieren und Deinstallieren.

Paketverwaltungen werden als einer der großen Vorteile und Erfolge der unixoiden Betriebssysteme beschrieben, z. B. von Ian Murdock als „the single biggest advancement Linux has brought to the industry“.[2] Eine weitere typische Eigenschaft dieser ist jedoch auch die Verwischung der Grenzen zwischen Anwendungen und Betriebssystem durch die integrierte Verwaltung durch die Distributionen, d. h. das Betriebssystem agiert nicht als Plattform für Anwendungen, sondern beinhaltet diese.[3][4] Einerseits ist mit einem systemweiten Paketmanagement ein konfliktfreier Betrieb (ohne Bibliotheks-Konflikte) des Betriebssystems mit den Applikationen sichergestellt, andererseits wird eine direkte Verteilung von Anwendungssoftware durch den Entwickler an die Kunden schwieriger[5][6], sowie auch die Erstellung von Portable Software.[7] Durch inkompatible Pakete zwischen den Distributionen ist die Verbreitung von Software ebenfalls erschwert[8]; hiergegen versucht die LSB Standards zu definieren und verbreiten, bis jetzt jedoch nur mit begrenztem Erfolg.[9]

Auch kann es bei der Softwareverwaltung zu Konflikten kommen (die aber erkannt werden): Sind in den Paketen A und B teilweise gleiche Dateien enthalten, können nicht beide Pakete gleichzeitig installiert werden. Auf einem System ohne eine Paketverwaltung würden die Dateien überschrieben, was zu Problemen bei der Ausführung der betroffenen Software führen kann. Ebenso kann es passieren, dass bei einer Aktualisierung des Pakets X auch die Aktualisierung des Pakets Y gefordert wird, Paket Z aber fordert, dass Y die Version beibehält – eine Aktualisierung ist dann nicht möglich. Es zeichnet eine gute Paketverwaltung aus, Konflikte und Abhängigkeiten richtig zu berechnen und zur bestmöglichen Lösung zu kommen, also die richtigen Pakete zu aktualisieren, wenn nötig veraltete Pakete zu löschen und den Konflikt so bestmöglich zu lösen.

Quellenbasierte Distributionen wie etwa Gentoo Linux begegnen diesen Problemen so: Hier werden die Software-Pakete erst auf dem Zielrechner kompiliert. Dabei ist es auch möglich, die Komponenten und damit die Abhängigkeiten eines Paketes anzupassen. Sollte ein bereits installiertes Paket nicht die benötigten Bibliotheken für ein neues Paket installiert haben, wird es kurzerhand neu übersetzt und erneut installiert.

## KONZEPT ANHAND „UBUNTU“

Im Gegensatz zu anderen bekannten Betriebssystemen ist Ubuntu mehr als nur eine Plattform, um Anwendungen auszuführen. Ubuntu ist eine GNU/Linux-Distribution. Das bedeutet, dass neben dem Betriebssystem an sich eine riesige Sammlung an Programmen zur Verfügung steht. So können zusätzliche Software, Anwendungen und Treiber über wenige Mausklicks aus einer sicheren Quelle installiert werden, ohne dass man vorher auf der Homepage des Herstellers eine Installationsdatei (Setup) herunterladen muss.

Ubuntu nutzt konsequent APT (Advanced Packaging Tool) bzw. dpkg für die Paketverwaltung. Somit kann Software in Paketen zuverlässig installiert und wieder deinstalliert werden. Von APT selber bekommt man als Benutzer wenig mit, sondern sieht nur die verschiedenen Oberflächen, welche auf APT aufsetzen. Welche man benutzt, ist eigentlich egal, denn alle steuern am Ende nur APT an.

Alle Desktop-Umgebungen beinhalten jeweils Programme, die umfangreiche Funktionen zur Verwaltung der Pakete bieten. Auch auf der Kommandozeile lassen sich die Funktionen von APT nutzen, was aber eher Fortgeschrittenen zu empfehlen ist.

<sup>23</sup> Vgl. <https://de.wikipedia.org/wiki/Paketverwaltung>, aufgerufen am 25.12.2017

Seit Ende 2014 gibt es mit snap-Paketen eine weitere Möglichkeit, Software parallel und konfliktfrei zur normalen Paketverwaltung zu installieren. Alle Pakete, die zum System und zum Desktop gehören, werden aber weiterhin (ausschließlich) via apt installiert.

## WEITERE PAKETVERWALTUNGEN

Durch die große Bandbreite an Linux-Distributionen gibt es auch verschiedene Verwaltungen. Für Linux Mint ist für uns hauptsächlich APT interessant, trotzdem hier eine Auflistung weiterer:

- Debian Package Manager + Advanced Packaging Tool (**APT**) – für deb- oder RPM-basierte Linux-Distributionen
- autopackage – ein distributionsunabhängiges Softwareinstallationssystem[15]
- Click'n'Run (CnR) – für Linspire (basierend auf Debians APT)
- NuGet – für Visual Studio
- gem – für Ruby-Pakete via RubyGems
- iPKG/opkg – eine Paketverwaltung speziell für Computer mit wenig Ressourcen, wird z. B. von OpenWrt verwendet
- Pacman von Arch Linux
- Perl Package Manager – für Active Perl Module
- Paketverwaltung PiSi der Linux-Distributionen Pardus und Solus
- pkgsrc für NetBSD, DragonFly BSD und viele weitere Unix-artige Betriebssysteme
- emerge und Portage – für Gentoo
- Ports-System unter BSD-Distributionen (z. B. FreeBSD, OpenBSD)
- Smart Package Manager – dieses Werkzeug greift auf bestehende Werkzeuge (rpm, deb, ...) zurück, und kann so auf vielen Distributionen eine einheitliche Oberfläche bieten
- Up2date – RPM-basiert
- User Red Hat Package Manager von Mandriva Linux – RPM-basiert
- **rpm und YUM** – für die Linux-Distributionen Fedora und Yellow Dog Linux
- xbps – für Void Linux
- **Zypper/YaST** unter OpenSUSE
- Bower – für clientseitige Webentwicklung
- Chocolatey - ist ein freier Paketmanager für Windows-Betriebssysteme

## PAKETE & PAKETQUELLEN

Programme werden bei Ubuntu in Paketform installiert. Diese Pakete werden durch Paketquellen (Repositories) im Internet zur Verfügung gestellt.

Besteht jedoch keine oder nur eine langsame Internetverbindung, kann man über die Einrichtung einer lokalen Paketquelle nachdenken. Lokale Paketquellen sind Paketquellen auf Datenträgern (CD/DVD, USB-Stick, externer Festplatte) oder im LAN.

Die eigentliche "Paketquelle" (auch Paketdepot genannt) ist eine Internetadresse, z.B. [archive.ubuntu.com/ubuntu](http://archive.ubuntu.com/ubuntu) oder ein Medium mit gespeicherten Paketen. Zudem wird nach "Archiven" unterschieden, etwa xenial oder xenial-security. Diese Archive sind jeweils nochmal unterteilt in einzelne "Komponenten".

Komponenten werden manchmal auch fälschlicherweise Sektionen genannt. Im Zusammenhang mit der Paketverwaltung versteht man unter "Sektionen" Kategorien, in denen Programme nach ihrer Art einsortiert



werden, etwa "Netzwerk", "GNOME", "Mathematik", usw. Am Beispiel von Ubuntu –es handelt sich bei Linux Mint um ein ähnliches Konzept- erläutern wir hier die unterschiedlichen Archive:

---

#### XENIAL

Das ist das "Haupt-Archiv" der jeweiligen Ubuntu-Version. Sobald die jeweilige Ubuntu-Version fertig ist (Final Status) und veröffentlicht wurde, wird an diesem Archiv nichts mehr geändert. Es kommen weder neue Pakete hinzu noch werden die vorhandenen Pakete in einer neuen Version zur Verfügung gestellt, geschweige denn einzelne Pakete aus dem Archiv entfernt. Selbst Sicherheits-Aktualisierungen oder Fehlerkorrekturen werden hier nach der Veröffentlichung nicht mehr eingepflegt. Dies geschieht über eigene Archive (siehe die folgenden Abschnitte).

---

#### XENIAL-SECURITY (WICHTIGE SICHERHEITSAKTUALISIERUNGEN)

Da aber Aktualisierungen unvermeidlich oder gar zwingend notwendig sind, gibt es ein "Archiv mit Sicherheitsaktualisierungen". Die Pakete aus diesem Archiv enthalten keine Versionen mit neuen Funktionen. Es handelt sich lediglich um Versionen aus dem Hauptarchiv, bei denen schwerwiegende Fehler und Sicherheitslücken korrigiert wurden. Diese Pakete nennt man deshalb im Englischen auch patched, also fehlerbereinigt. Neben dem Hauptarchiv sollte dieses Archiv definitiv verwendet werden, um den sicheren Betrieb des Systems zu gewährleisten.

---

#### XENIAL-UPDATES (EMPFOHLENE AKTUALISIERUNGEN)

In diesem "Archiv für aktualisierte Pakete" befinden sich ebenfalls lediglich fehlerbereinigte Pakete. Hier sind aber die behobenen Fehler weder gravierend noch sicherheitsrelevant, daher ist die Installation nicht zwingend erforderlich, schadet aber auch nicht.

---

#### XENIAL-BACKPORTS (NICHT UNTERSTÜTZTE AKTUALISIERUNGEN)

Die Backports enthalten neuere Programmversionen, die aus der aktuellen Entwicklerversion zurückportiert werden. Sie sind weder für sich genommen noch im Zusammenspiel mit dem Rest des Systems so gut getestet wie die meist nur geringfügig älteren Versionen, die ohne diese Paketquelle installiert sind. Dieses Archiv ist seit Ubuntu 11.10 standardmäßig aktiviert, jedoch nur für die manuelle Installation von Paketen konfiguriert. D.h. ein Paket aus den Backports muss explizit ausgewählt werden, ansonsten werden die Haupt-Archive verwendet.

#### NÜTZLICHE TOOLS

## TEXTBASIERTE ANWENDUNG

Hier eine kompakte Übersicht von häufig verwendeten Linux Befehlen in der Praxis:

### GRUNDLAGEN

#### AUFLISTEN

```
| ls -[OPTION]
```

Dieser Befehl listet im aktuellen Verzeichnis alle Dateien auf:

<b>ls -a</b>	Listet alles auf, auch versteckte Dateien
<b>ls -B</b>	Listet keine Einträge auf, welche auf "~" enden
<b>ls -d</b>	Listet Verzeichnisse auf
<b>ls -f</b>	Listet nur Files auf
<b>ls -s</b>	Listet alles auf, aber nach Größe
<b>ls -t</b>	Listet kürzlich geänderte Dateien auf
<b>ls -l</b>	Listet Ergebnisse zeilenweise auf

#### AUSGEBEN

```
| cat -[OPTION]
```

**Ohne Optionen:** *gibt den Inhalt der Datei in der Konsole aus*

```
| tail -[OPTION] <Name>
```

<b>tail &lt;name&gt;</b>	Zeigt das Ende der Datei an
<b>tail -f &lt;name&gt;</b>	Zeigt die zuletzt geschriebenen Zeilen an

#### HILFE ANZEIGEN

```
| man <Befehl>
```

```
| <Befehl> --help
```

```
| info <Befehl>
```

#### WECHSELN ZWISCHEN VERZEICHNISSEN

```
| cd [OPTION] [DATEI/VERZEICHNIS]
```

<b>cd /&lt;Verzeichnis&gt;</b>	Wenn man vom obersten Verzeichnis hinunterwechselt
<b>cd ..</b>	ins übergeordnete Verzeichnis

---

## SONSTIGES

```
| sudo halt
```

Der Computer wird durch diesen Befehl, nach Eingabe des Passwortes, (radikal) heruntergefahren.

## DATEIMANAGEMENT

---

### ANLEGEN VON VERZEICHNISSEN/DATEIEN

```
| touch
```

<b>touch &lt;DATEINAME&gt;</b>	Damit wird eine leere Datei erstellt.
--------------------------------	---------------------------------------

```
| mkdir
```

<b>mkdir &lt;ORDNERNAME&gt;</b>	Damit wird ein leeres Verzeichnis erstellt.
---------------------------------	---

---

### LÖSCHEN VON VERZEICHNISSEN/DATEIEN

```
| Rm -[OPTION]
```

<b>rm &lt;DATEINAME&gt;</b>	Damit wird eine einzelne Datei gelöscht.
-----------------------------	--

<b>rm -rf &lt;ORDNERNAME&gt;</b>	Damit wird ein ganzer Ordner (rekursiv) gelöscht.
----------------------------------	---

---

### ANZEIGEN VON VERZEICHNISSEN/DATEIEN

```
| less <Dateiname>
```

<b>less &lt;Dateiname&gt;</b>	Datei wird in Konsole ausgegeben und Navigation mit Pfeiltasten möglich
-------------------------------	---

---

### KOPIEREN VON VERZEICHNISSEN/DATEIEN

```
| cp -[OPTION]
```

<b>cp &lt;DATEINAME&gt;</b>	Damit wird eine einzelne Datei kopiert.
-----------------------------	---

<b>cp -r &lt;ORDNERNAME&gt;</b>	Damit wird ein ganzer Ordner (rekursiv) kopiert.
---------------------------------	--

---

### VERSCHIEBEN VON VERZEICHNISSEN/DATEIEN

```
| mv (dir)
```

<b>mv &lt;DATEINAME&gt; &lt;ENDVERZEICHNIS&gt;/[DATEINAME]</b>	Damit wird eine einzelne Datei im angegebenen Ordner verschoben
--	---

```
mvdir <ORDNERNAME> <ENDVERZEICHNIS>/[ORDNERNAME]
```

Damit wird ein ganzes Verzeichnis im angegebenen Ordner verschoben.

## SOFTLINK

d.h. Verknüpfung zur Datei. Damit kann man auch ganze Ordner verknüpfen.

```
ln -s <DATEINAME> <ENDVERZEICHNIS>/[DATEINAME]
```

Damit wird eine Verknüpfung dieser Datei im angegebenen Ordner erstellt.

```
ln -s <ORDNERNAME> <ENDVERZEICHNIS>/[ORDNERNAME]
```

Damit wird eine Verknüpfung eines ganzen Ordners in angegebenen Ordner erstellt.

## HARDLINK

d.h. identische Kopie einer Datei. **VORSICHT!** Man kann *keine Ordner Hardlinken!*

```
ln <DATEINAME> <ENDVERZEICHNIS>/[DATEINAME]
```

Damit wird eine identische Kopie dieser Datei im angegebenen Ordner erstellt.

## SUCHEN UND FINDEN

### FIND

```
| find -[OPTION] [weitere OPTION]
```

<b>find -ctime</b>	Zeit der Veränderung des Statuses
<b>find -type f</b>	Nur Files
<b>find -atime</b>	Zeit des Zugriffs
<b>find -mtime</b>	Zeit der Veränderung des Inhaltes

### LOCATE

```
| locate <name>
```

**Benötigt DATENBANK!!!** Manuelles Updaten mit:

```
| updatedb
```

### EGREP

```
| egrep -[OPTION] <name>
```

---

## SUCHEN NACH PROGRAMMEN

```
| which <Befehl/ Programm>
```

<b>which cd</b>	Ähnlich wie whereis aber ohne MAN -> man kann Befehle suchen
-----------------	--

## RECHTEMANAGEMENT

---

### ANZEIGEN VON RECHTEN

```
| id
```

<b>id</b>	Zeigt User und Gruppenmitgliedschaften des aktuellen Users an
-----------	---

```
| groups
```

<b>groups</b>	Zeigt Gruppenmitgliedschaften des aktuellen Users an
---------------	--

---

### BENUTZER WECHSELN

```
| su <Username>
```

<b>su</b>	Wechsel zu Root-Benutzer
<b>su &lt;name&gt;</b>	Wechsel zum angegebenen Benutzer
<b>su -username &lt;name&gt;</b>	Wie oben, ABER Umgebungsvariablen werden mitübernommen

---

### USER-RECHTE

---

### GROUP-RECHTE

---

### OTHERS-RECHTE

---

### ADMINISTRATOREN-RECHTE

## MOUNTEN

---

### MANUELLES MOUNTEN

```
| sudo mount <DEVICE> <MOUNT-POINT>
```

<b>mount /dev/sda1</b>	Mountet die Partition 1 von der Festplatte sda (wenn der Eintrag in fstab vorhanden ist!)
<b>mount /dev/sda1 /mnt</b>	Mountet die Partition 1 von der Festplatte sda auf /mnt (geht auch ohne Eintrag in fstab)

```
mount -t nfs 10.0.xx.2:<FREIG. ORDNER> <ZIELORDNER>
```

Mountet den freigegebenen Ordner von einem anderen Rechner in den gewünschten Zielordner

## UNMOUNT

```
| sudo umount <DEVICE> <MOUNT-POINT>
```

```
sudo umount /dev/sda1
```

Unmountet die Partition 1 von der Festplatte sda

## PROZESSVERWALTUNG

### HERAUSFINDEN DER PROZESS-ID...

#### ...DURCH ID

```
| ps aux | egrep '<ID>'
```

```
ps aux
```

Listet alle laufende Prozesse auf

```
ps aux | egrep <ID>
```

Sucht <ID> bei den laufenden Prozessen und es zeigt sie an, wenn verfügbar

Oder man kann alle laufenden Prozesse anzeigen lassen:

```
| top
```

#### ...DURCH NAMEN

```
| ps aux | egrep '<Name>'
```

## VORDER- UND HINTERGRUNDAUSFÜHRUNG

### MANAGEN VON PROZESSEN

```
| sudo service <name> start
```

```
| sudo service <name> stop
```

```
| sudo service <name> restart
```

---

## ÜBERWACHUNG VON PROZESSEN

```
| ps aux
```

---

## BEENDEN VON PROZESSEN

```
| kill -[OPTION] <Prozess-ID>
```

<b>kill -9 &lt;Prozess-ID&gt;</b>	Fordert den Prozess auf, sich selbst zu beenden. Eventuell werden Änderungen im geöffneten Programm gespeichert
<b>kill -15 &lt;Prozess-ID&gt;</b>	Der Prozess wird radikal beendet Vergleichbar mit sudo halt

## BEST PRACTICE

## INFORMATIONEN ANZEIGEN

## HILFE ANZEIGEN

Man kann sich nie alle Optionen für einen Befehl merken. Für alle Befehle in Linux gibt es deswegen eine Hilfeseite bzw. ein Manual. Dies ist eine komplette Beschreibung des Befehls. Aufrufen kann man das wie folgt:

```
| man <Befehl>
```

Genauso kann man aber auch ein `/?` oder `--help` nach dem Befehl als Option angeben. Dies zeigt eine kurze Zusammenfassung der Optionen an:

```
| <Befehl> /? oder --help oder -? oder /help
```

## INFORMATIONEN ZUM BETRIEBSSYSTEME, ETC.

Das aktuelle, in dem man sich befindet, Verzeichnis kann man mit **print working directory** anzeigen bzw. ausgeben lassen:

```
| pwd
```

Die aktuelle Linux Kernel Version und weitere Informationen zum Betriebssystem kann man mit `uname` ausgeben lassen:

```
| uname -[OPTION]
```

<b>uname -a /--all</b>	alle Informationen
<b>uname -s /--kernel-name</b>	Namen des Kernels
<b>uname -n /--nodename</b>	Netzwerknamen der Maschine (entspricht dem Befehl <code>hostname</code> )
<b>uname -r /--release</b>	Release-Nummer des Betriebssystems
<b>uname -v /--kernel-version</b>	Kernel-Version
<b>uname -m /--machine</b>	Maschinen Architektur (entspricht dem Befehl <code>arch</code> )
<b>uname -p /--prozessor</b>	Prozessor Typ
<b>uname -i /--hardware-platform</b>	Hardware Plattform

Mit `history` kann man einen Verlauf aller vergangenen Eingaben in den Terminal anzeigen lassen.

```
| history
```

Die Festplattenbelegung kann man mit `df` (disk free) anzeigen. Gegenstück hierzu ist `du` (disk used).

```
| df -[OPTION]
```

<b>-a</b>	<b>--all</b>	Einbeziehung aller Dateisystemen auch von 0 Blockgrößen
<b>-B</b>	<b>--block-size=Größe</b>	Angabe der zu benutzenden Blockgröße
<b>-h</b>	<b>--human-readable</b>	Gibt die Größe in einem vernünftigen (menschlichen) Format an z.B. 1K 234M 2G
<b>-i</b>	<b>--inodes</b>	Auflistung der INode-Information statt der Block-Benutzung
<b>-k</b>		Festlegung der Blockgröße auf 1k



<b>-l</b>	<b>--local</b>	Ausgabe auf lokale Dateisysteme begrenzen
<b>-t</b>	<b>--type=TYP</b>	Ausgabe auf Dateisysteme des Typs TYP begrenzen
<b>-T</b>	<b>--print-type</b>	Dateisystemtyp ausgeben
<b>-x</b>	<b>--exclude-type=TYP</b>	Ausgabe aller Dateisysteme außer vom Typ <b>TYP</b>

## .BASHRC

Die `bashrc` ist eine Konfigurations-Datei für die Bash, welche bei jedem Öffnen eines Terminals<sup>[1]</sup> mitgeladen wird. In ihr kann man Abkürzungen für Befehle (also ein alias) eintragen, man verändert hier das Aussehen des Prompts oder man kann zusätzliche Skripte einfügen, welche einem das Arbeiten mit der Shell erleichtern.

Man hat grundsätzlich zwei `bashrc`-Dateien. Eine findet man unter `/etc/bash.bashrc`. Dies ist die globale `bashrc`, was bedeutet, dass Änderungen, die man in dieser vornimmt, Auswirkungen auf die Bash von sämtlichen Benutzern des Systems haben. Die zweite `bashrc` findet man versteckt im Homeverzeichnis unter `.bashrc`. In dieser kann jeder Benutzer Änderungen nur für sich vornehmen und in dieser sollte man grundsätzlich seine Änderungen auch eintragen, außer man hat einen gewichtigen Grund, dass sie für alle Benutzer gelten sollen. Die globale `bash_profile` befindet sich in `/etc/profile` und die des Benutzers unter `~/.bash_profile`. Letztere existiert meist noch nicht und muss erst von Hand erstellt werden.

## COLORFUL TERMINAL

Es besteht auch die Möglichkeit den Terminal einzufärben. Order bzw. Dateien oder Links werden färbig dargestellt. Hierzu muss man die jeweilige `.bashrc` des Users editieren und die Zeile `#force_color_prompt=yes` auskommentieren, dh. das Raute-Zeichen vor der Zeile entfernen.

```
# uncomment for a colored prompt, if the terminal has the capability;
turned
# off by default to not distract the user: the focus in a terminal
window
# should be on the output of commands, not on the prompt
force_color_prompt=yes

if [ -n "$force_color_prompt" ]; then [...]
```

## ALIAS FÜR BEFEHLE

Weiters ist es durchaus nützlich für häufig genutzte commands sich Aliases bzw. Shortcuts anzulegen. Dies ist ebenfalls in der jeweiligen `.bashrc` des Users zu tun. Grundsätzlich gibt es dort schon vorgefertigt für die Verzeichnisauflistung Aliases, die man nur mehr auskommentieren muss:

```
# some more ls aliases
#alias ll='ls -l'
#alias la='ls -Al'
#alias l='ls -CF'
```

Nach dem Schema kann man auch selbst weitere Aliases anlegen. Nach den auskommentieren der fett markierten Zeilen kann man anstatt `ls -l` nur mehr `ll` eingeben und bekommt die gleiche Ausgabe.

## UPDATES AUTOMATISCH INSTALLIEREN (CRONTAB)

Der Cron-Daemon ist ein Dienst, der automatisch Skripte und Programme zu vorgegebenen Zeiten starten kann. Der auszuführende Befehl wird in einer Tabelle, der "crontab", gespeichert. Es gibt eine systemweite Datei /etc/crontab, die nur mit Root-Rechten bearbeitet werden kann. Zusätzlich kann jeder Benutzer eine eigene Crontab erstellen, die man dann im Verzeichnis /var/spool/cron/crontabs/ findet. Vergleichbar mit der Aufgabenplanung unter Windows.

### CRONTAB ANLEGEN

Aus historischen Gründen wird eine Crontab mit einem Kommandozeileneditor bearbeitet. Hierfür gibt man den Befehl

```
| crontab -e
```

ein. Wenn dies zuvor noch nie gemacht wurde, wird man der Festlegung für den zu verwendenden Editor gefragt. Am einfachsten ist hierbei nano zu bedienen. Anschließend öffnet sich der Editor mit dem config-file von cron. Einträge in diesem File sind wie folgt aufgebaut:

```
| *      *      *      *      *      [auszuführender Befehl]
|      -      -      -      -      -
|      |      |      |      |
|      |      |      |      +----- Wochentag (0 - 7) (Sonntag ist 0 und 7)
|      |      |      +----- Monat (1 - 12)
|      |      +----- Tag (1 - 31)
|      +----- Stunde (0 - 23)
+----- Minute (0 - 59; oder Namen, siehe unten)
```

In einer Zeile kommen zuerst die fünf Felder für die Zeiten (Minute, Stunde, Tag, Monat, Wochentage) und danach der auszuführende Befehl. Hier noch eine kleine Hilfestellung zu den Zeiten:

String	Bedeutung	cron-Schreibweise
@reboot	einmalig, beim Start	
@daily	einmal pro Tag	0 0 * * *
@midnight	einmal pro Tag	0 0 * * *
@hourly	einmal pro Stunde	0 * * * *
@weekly	einmal pro Woche	0 0 * * 0
@monthly	einmal im Monat	0 0 1 * *
@annually	einmal im Jahr	0 0 1 1 *
@yearly	einmal im Jahr	0 0 1 1 *

Um wieder auf unser Fallbeispiel zurück zu kommen: Wir wollen, dass unser Linux Gerät automatisch täglich um Punkt Mitternacht Updates macht. Wir erinnern uns zurück, dass Updates über den Terminal mit apt-get update && apt-get upgrade gemacht werden. Bei upgrade fügen wir noch den Parameter y hinzu, sodass keine Benutzereingabe erforderlich ist. Dieser Cronjob muss von root ausgeführt werden.

```
| 0 0 * * * apt-get update && apt-get upgrade -y
```

## BYOBU

Byobu ist ein kleiner, effektiver und text basierender window manager, welcher auf GNU Screen basiert. Dieses Tool erlaubt es, schnell zwischen verschiedenen Screens hin und her zu wechseln. Der Vorteil liegt darin, dass dies über eine einzige SSH Verbindung funktioniert. Weiters blendet byobu im unteren Bildschirmrand alle wichtigen Informationen zum System ein. Bevor es verwendet werden kann, muss es installiert werden, da es nicht standardmäßig vorhanden ist:

```
| apt-get install byobu
```

Anschließend muss es noch für jeden User extra aktiviert werden, der es verwenden möchte.

```
| byobu-enable
```

Die Verwendung gestaltet sich grundsätzlich immer durch die Funktionsaufruf mit der Tastenkombination [Strg] + [A]. Anschließend können dann folgende Parameter eingegeben werden:

[C]	Erzeugt ein neues Fenster
[N]	Wechselt zum nächsten Fenster
[D]	Schließt das aktuelle Fenster

Weitere Informationen findest du unter [www.byobu.co](http://www.byobu.co)

## DATEIMANAGEMENT

Nachdem die Befehle schon ausführlich erläutert wurden im Befehlsverzeichnis, hier nur eine kurze Zusammenfassung bzw. wie man schnell Dateien anlegen und befüllen kann.

### SCHNELLES ERSTELLEN

Wie schon erwähnt kann man mit dem Befehl `touch` Dateien aller Art schnell anlegen. Wenn man zB. eine Ausgabe in eine Datei kopieren will, kann diese mit `cat` ausgeben und durch zwei Spitzklammern in die Datei umleiten:

```
| touch a.txt
| cat somedata.txt >> a.txt
```

Selbiges kann auch mit Ausgaben von Befehlen oder simple von `echo` gemacht werden. Bei dieser Variante erspart man sich sogar das Anlegen der Datei:

```
| echo "hello world" >> new.txt
```

### SCHNELLES UMBENENNEN

Eine Datei kann umbenannt werden, indem sie mit dem Befehl `mv` im gleichen Verzeichnis verschoben wird:

```
| mv old.txt new.txt
```

Wenn die alte Datei noch vorhanden sein soll, kann man auch den Befehl `cp` verwenden.

### SECURE COPY

Das Kommandozeilenwerkzeug `scp` funktioniert in etwa so wie das normale Unix-Kommando `cp`, nur dass es über Systemgrenzen hinweg funktioniert. Jedes Datei- oder Verzeichnisargument kann dabei optional, getrennt durch einen Doppelpunkt, durch einen vorangestellten Benutzer- bzw. Hostnamen ergänzt werden. Dabei werden

weggelassene Teile durch den aktuellen Benutzernamen, localhost bzw. das Homeverzeichnis (bzw. das aktuelle Verzeichnis) ergänzt, etwa so:

```
scp [user]@[server]:[Pfad][Datei] [user]@[server]:[Pfad][Datei]
scp one.txt tux@myserver.com:~/data/
scp tux@second.myserver.com:/home/tux/test.txt ./
```

## PAKTEINSTALLATION

Wie schon vorher in der Befehlszusammenfassung erläutert, kann man in Debian, Ubuntu und Linux Mint Programme mit apt-get über das Terminal Programme installieren. Wenn man sich nicht sicher ist wie das Paket genau heißt, kann man mit

```
apt-cache search [STRING]
```

mögliche Vorschläge zu dem eingegebenen Suchstring anzeigen lassen. Wenn man das Paket gefunden hat, welches man installieren möchte, tut man dies mit:

```
apt-get install [STRING]
```

## VIM



Viele Benutzer schrecken vor dem sehr praktischen Editor VIM wegen seiner etwas sperrigen Bedienung zurück. Wer sich aber erst einmal mit den verschiedenen Betriebsmodi vertraut gemacht hat, mag auf die hilfreichen Dienste des schlanken, aber fast beliebig erweiterbaren Editors nicht mehr verzichten. VIM ist eine Weiterentwicklung des Texteditors Vi, deshalb auch der Name als Abkürzung für "Vi IMproved". Es gibt Editoren, die anfangs wesentlich leichter zu bedienen sind, z.B. Pico oder Nano. Jedoch hat VIM einige Vorteile:

- Auf jedem POSIX-konformen System ist mindestens der Vorfahr des VIM, der Vi, als Editor vorhanden (wobei Vi sehr viel weniger Funktionen hat)
- Navigierbar ohne Cursor-Tasten (kann bei Tastaturproblemen in Remote-Sessions helfen)
- Superschnelle Bedienbarkeit
- Fast beliebig erweiterbar und konfigurierbar (Einstellungen, Macros, Mappings, Plugins ...)

Es empfiehlt sich für alle, die noch nicht mit `vim` gearbeitet haben, `vimtutor` zuerst sich anzuschauen.

## BEDIEHUNG

### SPEICHERN UND BEENDEN:

Befehl	Beschreibung
<b>:q</b>	Schließt das aktuelle Fenster. Falls dies das letzte Fenster ist und irgendein Buffer noch nicht gespeichert wurde, wird (je nicht gespeichertem Buffer) eine Rückfrage gestellt, ob die Änderungen verworfen werden sollen.
<b>:wq</b>	Speichert den aktuellen Buffer und schließt das Fenster. Falls dies das letzte Fenster ist und irgendein Buffer noch nicht gespeichert wurde, wird (je nicht gespeichertem Buffer) eine Rückfrage gestellt, ob die Änderungen verworfen werden sollen.
<b>:qa11</b>	Schließt VIM. Falls Buffer noch nicht gespeichert wurden, wird (je nicht gespeichertem Buffer) eine Rückfrage gestellt, ob die Änderungen verworfen werden sollen.

<b>:x</b>	Wie :wq.
<b>:q!</b>	Schließt das aktuelle Fenster. Falls dies das letzte Fenster ist und irgendein Buffer noch nicht gespeichert wurde, werden nicht gespeicherte Änderungen verworfen.
<b>:wn</b>	Speichert den aktuellen Buffer und wechselt zum nächsten Buffer.
<b>:qa11!</b>	Schließt VIM. Änderungen an noch nicht gespeicherten Buffern werden ohne Rückfrage verworfen.

## VIM MODI

Befehl	Beschreibung
<b>i</b>	Wechselt in den Eingabemodus links vom Cursor.
<b>a</b>	Wechselt in den Eingabemodus rechts vom Cursor.
<b>o</b>	Erstellt neue Zeile unterhalb der Aktuellen und wechselt in den Eingabemodus. Wenn autoindent aktiviert ist, wird die Einrückung automatisch vorgenommen.
<b>r</b>	Wechselt für ein Zeichen in den Ersetzungsmodus (Art des Eingabemodus, bei dem Zeichen überschrieben werden anstatt einzufügen). Wird beispielsweise rX eingegeben, wird das Zeichen unter dem Cursor mit dem Buchstaben X ersetzt.
<b>s</b>	Wechselt in den Eingabemodus und entfernt/ersetzt dabei das Zeichen unter dem Cursor. Wird beispielsweise sFOOBAR eingegeben, wird das Zeichen unter dem Cursor mit FOOBAR ersetzt, man bleibt weiterhin im Eingabemodus.
<b>cw</b>	Wechselt in den Eingabemodus und entfernt/ersetzt dabei das gesamte Wort ab und unter dem Cursor.
<b>I</b>	Wechselt in den Eingabemodus am Anfang der Zeile oder bei aktiviertem autoindent an den Anfang der Einrückung.
<b>A</b>	Wechselt in den Eingabemodus am Ende der Zeile.
<b>O</b>	Erstellt neue Zeile oberhalb der Aktuellen und wechselt in den Eingabemodus. Wenn autoindent aktiviert ist, wird die Einrückung automatisch vorgenommen.
<b>R</b>	Wechselt in den Ersetzungsmodus.
<b>S</b>	Wechselt in den Eingabemodus und entfernt/ersetzt dabei die gesamte aktuelle Zeile.
<b>cc</b>	Wie S.

## WEITERFÜHREND

Empfiehlt sich diese Website: <https://wiki.ubuntuusers.de/VIM/>

## TROUBLESHOOTING

## „TASKMANAGER“

Was tun, wenn ein Programm „abschmiert“ oder sich nicht beenden lässt? – Terminal aufmachen, sich alle laufenden Programme anzeigen lassen und nach dem Programmnamen filtern lassen: Der vertikale Strich nennt sich „Pipe“ und dient als Weiterleitung von einem Befehl in den nächsten:

```
| ps aux | egrep "[STRING]"
```

Mögliche Ausgabe:

```
www-data 30509 0.0 0.0 399452 10660 ? S 00:06 0:00
/usr/sbin/apache2 -k start
www-data 30510 0.0 0.0 399452 10660 ? S 00:06 0:00
/usr/sbin/apache2 -k start
www-data 30511 0.0 0.0 399452 10660 ? S 00:06 0:00
/usr/sbin/apache2 -k start
root 30516 0.0 0.0 96920 7640 ? Ss 00:06 0:00
/usr/sbin/cupsd -l
root 30517 0.0 0.0 258468 9008 ? Ssl 00:06 0:00
/usr/sbin/cups-browsed
```

Die Zeilenumbrüche sind zu vernachlässigen. Interessant ist hierbei nur die fett markierte Prozess-ID. Diese notieren und dann mit dem kill Befehl das Programm mit der ID XXXXX beenden: Die Option -9 bricht den Prozess radikal ab, -15 beendet ihn.

```
| kill [-9] [-15] [ID]
```

TIPP: Falls die GUI ebenfalls „abgeschmiert“ ist, kann man sich in den terminal modus „retten“ mit der Tastenkombination [Strg] + [Alt] + [F1].

## LOG-FILES

Protokolldateien oder "Logdateien" (engl. "log files") sind ein wichtiges Konzept eines Linux-Systems. Viele Programme, insbesondere alle Dienste und das System selbst, schreiben derartige Dateien. Dort kann man Fehler des Systems finden, Probleme von Diensten erkennen oder einfach nur verfolgen, was das System im Hintergrund macht.

Üblicherweise findet man die Protokolldateien im Verzeichnis /var/log. Manche Dienste, die z.B. mehrere Dateien schreiben, benutzen Unterverzeichnisse wie z.B. Samba mit /var/log/samba/.

Logdateien sind immer Textdateien, d.h. man kann sie mit jedem beliebigen Texteditor oder Dateibetrachter ansehen. Die Desktop-Umgebungen bringen Werkzeuge mit, um die Protokolle etwas komfortabler zu analysieren. Man kann aber auch das Terminal verwenden.

Mit dem Textbetrachter less oder tail kann man log-Files gut auslesen.

```
| less [FILE]
```

```
| tail [-f] [FILE]
```

```
| journalctl [OPTIONEN]
```

## DIENST NEU STARTEN

Ein Dienst ist ein Programm, das beim Start des Rechners automatisch ausgeführt wird und im Hintergrund darauf wartet, seine Aufgabe zu erledigen. Ein Dienst besitzt meist keine grafische Oberfläche und arbeitet ohne Interaktion des Benutzers. Die bekanntesten Dienste sind sicherlich Web-, Mail- oder Datenbank-Server. Aber auch die Hardwareerkennung oder das automatische Einbinden (Mounten) von z.B. USB-Sticks wird durch Dienste erledigt. Prinzipiell gibt es also zwei Arten von Diensten: interne für beim Systemstart relevante bzw. hardwarenahe Aufgaben und vom Benutzer nachinstallierte Dienste, zu denen in der Regel alle Serverdienste gehören.

Im Computer-Jargon werden Dienste traditionell auch als "Daemon" (im Sinne eines guten Geistes) bezeichnet. Der Buchstabe d findet sich daher häufig auch als letzter Buchstabe in der Programmbezeichnung wieder, so z.B. bei der Serverkomponente sshd von SSH.

Mit service bzw. systemctl kann man den status eines Dienstes erfahren, den Dienst starten und beenden:

```
| sudo service DIENSTNAME [OPTION]
```

<b>-start</b>	Startet den Dienst
<b>-stop</b>	Stoppt den Dienst
<b>-restart</b>	Startet den Dienst neu
<b>-status</b>	Zeigt den Status des Dienst an
<b>-reload</b>	Lädt den Dienst neu (kein Neustart)

```
| sudo systemctl [BEFEHL] [UNIT.service]
```

<b>-start</b>	Startet den Dienst
<b>-stop</b>	Stoppt den Dienst
<b>-restart</b>	Startet den Dienst neu
<b>-status</b>	Zeigt den Status des Dienst an
<b>-reload</b>	Lädt den Dienst neu (kein Neustart)

## NETZWERK

Anzeigen von der aktuellen IP-Konfiguration: Je nach LINUX Derviate kann der Befehl variieren. Hier eine kleine Aufstellung von möglichen Befehlen:

```
| ip [OPTION]
```

```
| ifconfig [OPTION]
```

```
| interfaces [INTERFACE-NAME] [OPTION]
```

Hiermit können alle Interfaces angezeigt bearbeitet oder deaktiviert werden. Mehr ist den man-Pages zu entnehmen.

Beobachtung von Netzwerkverkehr mit:

```
| tcpdump
```

```
| iftop
```

```
| netstat -tulpen
```



# IMPRESSUM



Herausgeber: **StV Lehramt TU Graz**

Verfasser: **Markus Wiener**

**Wir übernehmen keine Gewähr für die vollständige Richtigkeit und Aktualität dieses Dokuments.**

Fassung: **v1.1.0 vom 12.10.2018**