

# Linux für Einsteiger

DER EINSTEIGERKURS



Rene Furst

TUTKIT.COM DOZENT | EDITION 2017 | WIEN

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis.....</b>	<b>2</b>
<b>1. Vorwort .....</b>	<b>4</b>
<b>2. Der Autor.....</b>	<b>4</b>
<b>3. Der Linux Hintergrund.....</b>	<b>4</b>
3.1.1 Was versteht man unter Linux?.....	4
3.1.2 Linux Entstehung .....	4
3.1.3 Was ist ein Betriebssystem? .....	5
3.1.4 Was sind Linux Distributionen? .....	5
3.1.5 Welche Distribution ist die richtige für mich? .....	6
3.1.6 Freie und quelloffene Software.....	6
<b>4. Die Linux Installation .....</b>	<b>7</b>
<b>4.1 Linux Installation mit Virtual Box .....</b>	<b>7</b>
4.1.1 Virtual Box Downloaden und Installieren .....	8
4.1.2 Ubuntu Downloaden .....	8
4.1.3 Neue virtuelle Maschine erstellen.....	8
4.1.4 RAM Speicher definieren .....	8
4.1.5 Harddisk definieren .....	8
4.1.6 Maschine starten und Linux installieren.....	8
4.1.7 Linux Virtual Box Treiber .....	8
4.1.8 Netzwerk Übersichtstabelle Zugriffsmöglichkeiten in Virtual Box .....	9
4.1.9 SSH Server auf dem Virtuellen Server installieren.....	10
4.1.10 Mit dem Virtuellen Server vom Hauptsystem per SSH Verbinden .....	10
<b>4.2 Linux Installation bei DigitalOcean (Rootserver).....</b>	<b>11</b>
4.2.1 Server „Droplet“ erstellen bei DigitalOcean .....	11
4.2.2 Mit dem DigitalOcean via SSH Server Verbinden .....	12
<b>5. Die Linux Grundlagen.....</b>	<b>13</b>
5.1.1 Verzeichnisstruktur von Linux .....	13
<b>6. Die Linux Shell .....</b>	<b>15</b>
6.1.1 Die Bash (Shell) .....	15
6.1.2 Der Tab-Stopp Trick .....	15
6.1.3 Wildcards / Platzhalter ( * , ? , [ , [a-z], [0-9] ) .....	16
<b>7. Arbeiten mit Linux .....</b>	<b>16</b>
<b>7.1 Linux Basis Befehle.....</b>	<b>16</b>
7.1.1 Sudo und der Superuser (etwas Root User ausführen) .....	17
7.1.2 Inhalt eines Verzeichnisses ansehen mit „ls“ .....	17
7.1.3 Verzeichnis wechseln mit „cd“ .....	18
7.1.4 Wo befinde ich mich gerade mit „pwd“ .....	18
7.1.5 Zeige mir den Inhalt einer Datei mit „cat“ .....	19
7.1.6 Anzeige von Werten mit „echo“ .....	19
7.1.7 Dokumentationen zu Befehlen ansehen mit „man“ .....	20
7.1.8 Verlassen einer aktuellen Sitzung mit „exit“ .....	20
7.1.9 Den Bildschirminhalt löschen mit „clear“ .....	20
7.1.10 Die „help“ Option .....	20
<b>7.2 Arbeiten mit Dateien und Ordner.....</b>	<b>20</b>
7.2.1 Erstellen und Löschen eines neuen Ordners (mkdir, rmdir, rf) .....	20

7.2.2	Erstellen und Löschen einer neuen Datei .....	21
7.2.3	Kopieren von Dateien und Ordnern .....	21
7.2.4	Bewegen und Umbenennen von Dateien und Ordnern .....	22
7.2.5	Dateien Inhalte anzeigen (cat, more, less, head, tail, sort) .....	22
7.2.6	Nano Editor , eine Datei bearbeiten .....	22
7.2.7	Eine Datei ausführen .....	23
7.2.8	Archivieren von Dateien und Ordnern mit „tar“ .....	23
7.2.9	Komprimieren von Dateien und Ordnern mit „Zip, gzip“ .....	24
7.2.10	Dateigrößen Anzeigen mit „du“ .....	24
7.2.11	Dateien und Ordner suchen und finden mit „locate“ und „find“ .....	24
7.2.12	Filtern von der Ausgabe mit Pipes, Grep, Sort (   ) .....	25
7.2.13	Ausgabe in eine neue Datei speichern und in bestehenden anhängen .....	27
7.2.14	Dateien sicher und verschlüsselt kopieren über das Netzwerk mit SSH und SFTP ....	28
<b>7.3</b>	<b>Eigentum und Zugriffsrechte .....</b>	<b>29</b>
7.3.1	Besitzer von Dateien und Ordner mit chown .....	29
7.3.2	Zugriffsrechte auf die Dateien und Ordner .....	29
7.3.3	Zugriffsrechte unter Linux .....	30
7.3.4	Zugriffsrechte setzen mit chmod.....	30
7.3.5	Dateitypen erkennen .....	32
<b>7.4</b>	<b>Benutzer- und Gruppenverwaltung .....</b>	<b>32</b>
7.4.1	Der Root-User (Administrator) .....	32
7.4.2	Neuen Benutzer anlegen mit „adduser“ .....	33
7.4.3	Neuen Benutzer löschen mit „deluser“ .....	33
7.4.4	Gruppen anlegen mit „addgroup“ .....	33
7.4.5	Gruppen löschen mit „delgroup“ .....	34
<b>7.5</b>	<b>Verwaltung des Betriebssystems.....</b>	<b>34</b>
7.5.1	Cronjob - Wiederkehrende Aufgaben.....	34
7.5.2	Instanzen und Prozesse in Linux.....	35
7.5.3	Laufende Prozesse anzeigen mit „ps“ .....	35
7.5.4	Laufende Prozesse mit „top“ .....	36
7.5.5	Laufenden Prozesse beenden mit „kill“ .....	36
7.5.6	Offizielle Software Installieren mittels Paketmanager .....	37
7.5.7	Red Hat Derivat Software Installieren, Deinstallieren mit RPM .....	37
7.5.8	Debian Derivat Software mit APT .....	38
7.5.9	Pakete suchen mit „apt-cache“ .....	38
7.5.10	Pakete installieren mit „apt-get“ .....	39
<b>8.</b>	<b>Zusammenfassung .....</b>	<b>39</b>
8.1.1	Gratulation zum Linux für Anfänger Kurs .....	39
<b>8.2</b>	<b>Die wichtigsten Webseiten zum Thema.....</b>	<b>40</b>
8.2.1	Linux Distributionen und Tools.....	40
8.2.2	Die Besten und günstigsten Root und Virtuellen Server .....	40
8.2.3	Domain Registrierung und Webhosting .....	40
8.2.4	Wordpress .....	40
8.2.5	Videokurse.....	40
8.2.6	Autorenwebseite .....	40

## 1. Vorwort

Dieses E-Book ist eine Begleitung zum Tutkit Kurs „Linux für Einsteiger“. Es behandelt alle Themen die in diesem Kurs vorkommen und teilweise sogar darüber hinaus. Hier erfahren sie alles Wissenswerte zu Linux welche Ihnen den Einstieg in dieses fabelhafte Betriebssystem geben. Wir beginnen ganz von Anfang an und ich führe Sie an der Hand entlang durch dieses Buch. Angefangen von der Installation auf dem eigenen PC in einer Virtuellen Umgebung als auch in der Live Anwendung mit einem angemieteten Rootserver bei einem Provider. Am Ende des Buches und des Kurses gehen wir noch in eine Bonuslektion über wo ich Ihnen zeige wie Sie Ihre Linux Maschine als Datenbank und Webserver für Wordpress und andere Webseiten nutzen können. Wir gehen bis hin zur Wordpress Installation. Und nun lasst uns beginnen mit der Vorstellung des Autors als auch mit dem Linux Hintergrund.

## 2. Der Autor



Mein Name ist Rene und bin seit 2016 aktiver Dozent bei Tutkit. Mit über 17 Jahren Erfahrung in der IT Branche versuche ich nun mein Wissen über diese Plattform weiterzugeben und verfasse dabei auch Regelmäßig E-Books welche begleitend zum Kurs Anwendung finden sollen. Die Bücher sind jedoch so aufgebaut das man diese auch ohne Begleitkurs meistern und hoffentlich eine Menge dabei lernen

kann. Als IT Manager musste ich schon viele Aufgaben im Leben meistern, darunter zählen unter anderem die IT Sicherheit in Unternehmen, als auch die Programmierung kleinerer Programme welche die Produktivität der Mitarbeiter steigern können.

## 3. Der Linux Hintergrund

### 3.1.1 Was versteht man unter Linux?

Unter Linux versteht man ein freies Betriebssystem, welches wie Microsoft Windows und Apple Mac OS X die Schnittstelle zwischen der physischen Hardware und der eingesetzten Software bildet. Das meist verwendete Zweck des Linux Betriebssystem besteht derzeit darin als Server oder Smartphone Betriebssystem zu dienen. Mehr als 55% der Server weltweit nutzen Linux und etwa 82% nutzen Linux auf Ihrem Smartphone in form von Android. Im normalen Desktop Bereich spielt Linux mit unter 2% derzeit keine Rolle, weshalb es seine stärken auch im Serverbereich voll und ganz ausspielen kann.

### 3.1.2 Linux Entstehung

An dieser stelle möchte ich noch erwähnen nicht allzu sehr ins Detail zu gehen, sondern Kurzeinführung in die Entstehung von Linux zu geben.

Linus Torvalds ist der Initiator und die treibende Kraft bei der Entwicklung des Linux-Kernels.

1991 begann Linus mit der Entwicklung einer Terminal-Emulation um seinen eigenen Computer besser zu verstehen. Nach kurzer Zeit merkte er jedoch das sich das System immer mehr zu einem eigenständigen Betriebssystem entwickelte. Zuerst hatte Linus für sein Betriebssystem Linux eine proprietäre Lizenz festgelegt, welche die kommerzielle Nutzung verboten hat. Er merkte jedoch bald, dass der Fortschritt der Entwicklung dadurch behindert wurde. Er wollte den anderen Entwicklern mehr Freiraum geben und stellt darauf hin Linux unter die GNU GPL Lizenz. Dies war die Geburtsstunde des ersten freien Betriebssystems. Kurz darauf hin wuchs die Fangemeinde von Linux rasant an und blieb bis heute ungebrochen, bei den Entwicklern und Verfechtern freier Software.

### 3.1.3 Was ist ein Betriebssystem?

Ein Betriebssystem ist eine Zusammenstellung von Programmen welches die physischen Systemressourcen eines Computers wie RAM, Festplatte, Tastatur, Maus, Monitor u.v.m verwaltet. Betriebssysteme bestehen in der Regel auf einem Kernel (Kern), sowie speziellen Programmen die beim Start unterschiedliche aufgaben übernehmen.

Wenn ein Betriebssystem riesig, überladen und nicht optimiert ist, dann ist unter umständen das ganze System langsam, unabhängig davon welches Programm man auf Linux, Windows oder OS X startet. Das macht den entscheidenden unterschied aus welches Betriebssystem man ein setzen sollte.

Linux ist bekannt dafür das man es sehr schlank halten kann und nur das nötigste installieren kann. Somit können manche Linux Versionen in der Regel auch schon mit weniger als 100 MB auskommen, während Windows oder OS X mindestens 8.000 - 10.000 MB benötigen. Selbst Entwicklern von Microsoft und Apple gelingt es nach jeder weiteren Version Ihres Betriebssystems Verbesserungen zu implementieren, und somit die Systeme immer weiter zu verbessern und teilweise sogar massiv zu beschleunigen.

### 3.1.4 Was sind Linux Distributionen?

In den Anfängen von Linux gab es eine einzige Linux Version, die von Linus Torvalds entwickelt wurde. Unter einer Linux Distribution versteht man ein Komplettpaket das aufeinander abgestimmt ist für bestimmte aufgaben.

Eine Distribution besteht aus folgenden Bestandteilen:

- .) Betriebssystem (dem Linux-Kernel) mit seinen Grundfunktionen und Treibern
- .) Programmen und Oberfläche
- .) Quelltext bzw. Quellpaketen
- .) Dokumentation und Support

Die einzelnen Linux Distributionen können sich stark voneinander unterscheiden. Verschiedene Distributoren bieten spezielle Linux Distributionen für bestimmte aufgabengebiete an. So gibt es z.B. Linux Versionen wie Mint Linux, Ubuntu die sich speziell an Desktop (Heimbenutzer) wenden, als auch Versionen wie Debian, Red Hat für den Server Bereich. Darüber hinaus gibt es noch eigene Distributionen wie Tails für das anonyme Internet surfen, als auch Kali Linux für Sicherheitstests und Hacking. Sehr bekannt sind auch Distributionen für die PC Rettung, diese sind vorab ausgestattet mit sämtlichen

Virenskannern und Trojaner-Scanner um einen Windows PC wieder flott und Virenfrei zu bekommen.

Viele dieser Distributionen basieren auf anderen Distributionen sogenannten Derivate welche angepasst und erweitert werden. So besteht zum Beispiel Ubuntu auf einem Derivat von Debian, welche sich den selben unterbau bedienen. Somit ist es ein Kinderspiel wenn man mit Debian vertraut ist , sich auch gleich in Ubuntu zurecht zu finden. Insgesamt gibt es in etwa 5-6 Derivate welche sich unzählige Distributionen teilen.

Eine Übersicht über die Linux Distributionen findet man bei Wikipedia unter diesem Link.  
[https://de.wikipedia.org/wiki/Liste\\_von\\_Linux-Distributionen](https://de.wikipedia.org/wiki/Liste_von_Linux-Distributionen)

### 3.1.5 Welche Distribution ist die richtige für mich?

Dies ist immer eine schwierige Frage! Das hängt ganz von Ihrem Einsatzzweck und Vorlieben ab. Sie müssen sich auch im Klaren sein wo sie Ihre Linux Maschine einsetzen wollen. Benötigen sie z.B. einen Business Support? Kann Ihnen wer bei Problemen professionell helfen? Hierfür gibt es wieder eigene Enterprise Linux Distributionen mit professionellen Entwicklern dahinter. Ein gutes Beispiel hierfür ist Suse und Red Hat Enterprise, welches sich an Firmen richtet die größere Serverumgebungen mit Support benötigen. Es ist nicht unbedingt so das Red Hat Enterprise mehr kann als Debian oder andere Distributionen, jedoch sind die Strukturen und der Vertrieb dahinter auf „Enterprise Lösungen“ ausgelegt. Somit erhalten Sie auch professionellen Support. In diesem Buch befassen wir uns mit Debian Derivate wie z.B. Ubuntu. Vieles davon lässt sich jedoch auch auf andere Derivate umwälzen, da der Linux Kernel bei allen der gleiche ist. (Abgesehen von verschiedenen Versionsnummern)

### 3.1.6 Freie und quelloffene Software

In der Regel handelt es sich bei den meisten Linux Distributionen um eine freie Software. Doch was bedeutet freie und quelloffene Software eigentlich?

Freie Software bedeutet:

1. Die Freiheit, das Programm für jeden Zweck nutzen zu dürfen.
2. Die Freiheit, zu verstehen, wie das Programm funktioniert und wie man es seinen Ansprüchen anpassen kann.
3. Die Freiheit, Kopien weiterzuverbreiten, so dass man anderen damit helfen kann.
4. Die Freiheit, das Programm zu verbessern und die Verbesserungen allen öffentlich zur Verfügung zu stellen.

Quelloffene Software bedeutet aus zentralen Überzeugungen:

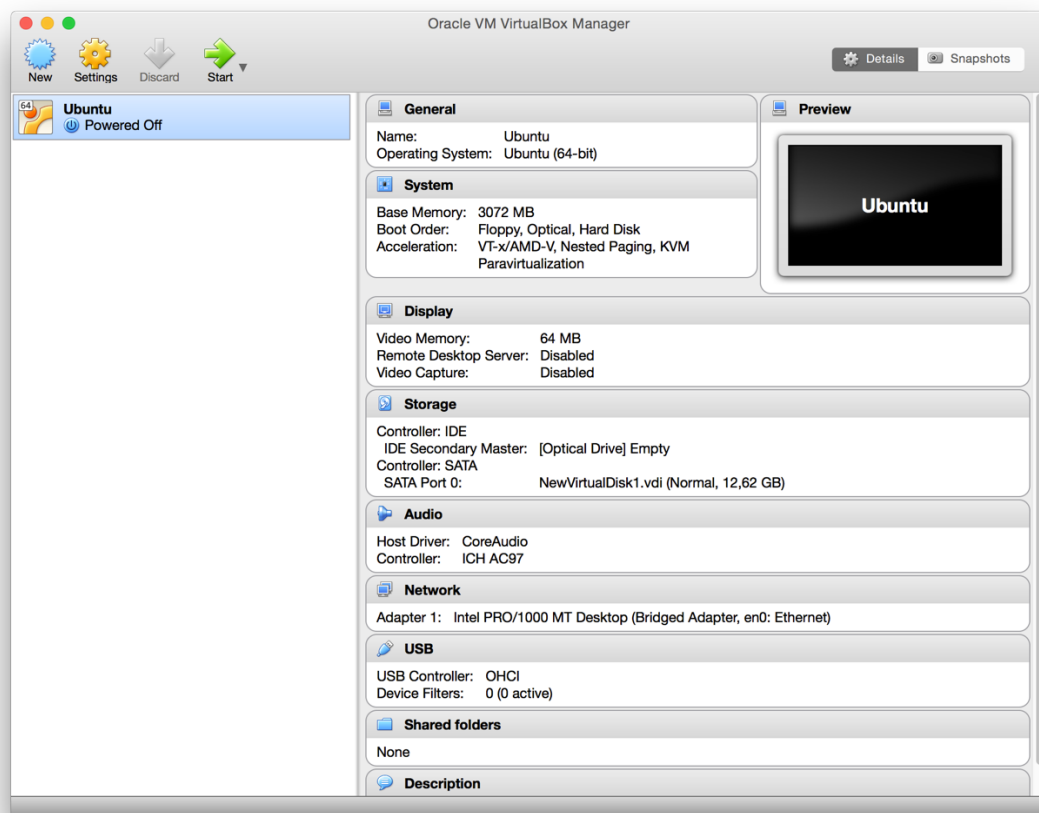
1. Jeder Benutzer eines Computers sollte seine Programme für jeden Zweck einsetzen, kopieren, weitergeben können oder Änderungen und Verbesserungen durchführen zu dürfen ohne gleich Lizenzgebühren bezahlen zu müssen.
2. Jeder Benutzer sollte auch die Möglichkeit haben seine Programme in einer Sprache seiner Wahl benutzen zu können.

3. Auch im Falle einer Behinderung sollte jeder Benutzer die Möglichkeit haben seine Programme anpassen zu dürfen.

Das sind Überzeugungen aus der Open Source Initiative welche quelloffene und freie Software beschreibt. Worunter sich auch Linux zählt und diese denken einer großen Community weitergibt.

## 4. Die Linux Installation

### 4.1 Linux Installation mit Virtual Box



Jetzt erstellen wir auf Ihrem PC / MAC einen Virtuellen Linux Server für eine Testumgebung. Hierfür benötigen wir ca. 20GB freien Festplattenspeicher und 2-4 GB RAM.

#### **WICHTIG:**

Sollten Sie lieber gleich mit einem Rootserver arbeiten wollen, welchen Sie dann Live einsetzen wollen, dann überspringen sie dieses Kapitel und fahren Sie beim Kapitel DigitalOcean fort.

#### 4.1.1 Virtual Box Downloaden und Installieren

Zuerst benötigen wir ein Programm welches einen Virtuellen PC simulieren kann. Hierfür eignet sich Virtual Box bestens, da es kostenlos ist und recht gut funktioniert.

Gehen Sie auf die Webseite von Virtual Box und Downloaden sie die neueste Version:  
<https://www.virtualbox.org>

Nach der Installation benötigen wir noch eine Linux CD/DVD bzw. Image Datei (ISO).

#### 4.1.2 Ubuntu Downloaden

Um an eine Image Datei von Linux zu gelangen besuchen Sie ganz einfach die Webseite der gewünschten Distribution. In diesem fall Downloaden wir die aktuellste Version von Ubuntu.

Besuchen Sie die Ubuntu Webseite und Laden Sie die ISO Datei herunter, welche ca. 1.5 GB groß ist.

<http://www.ubuntu.com>

#### 4.1.3 Neue virtuelle Maschine erstellen

Erstellen Sie nun eine neue Virtual Maschine indem Sie bei Virtual Box auf „New“ klicken. Wählen Sie dann einen Namen aus welche die Maschine erhalten soll und als Type & Version nutzen Sie „Linux“ und „Ubuntu (64-bit)“.

#### 4.1.4 RAM Speicher definieren

Anschließend wählen Sie wieviel Speicher Sie der Virtuellen Maschine zuteilen wollen. Hierfür empfiehlt es sich je nachdem wieviel RAM sie in Ihrem PC/Mac haben ca. 1/4 zu nutzen. Als Beispiel bei 8 GB RAM in Ihrem PC nutzen Sie 2 GB RAM in der virtuellen Maschine.

#### 4.1.5 Harddisk definieren

Da wir das Ubuntu in die virtuelle Maschine installieren wollen, benötigen wir eine virtuelle Harddisk. Hierfür sollten wir mindestens 12 GB (besser 20 GB) definieren. Wir klicken also auf „Create a virtual hard disk now“ und wählen dann VDI und „Dynamically allocated“ aus. Dies bedeutet das Virtual Box noch nicht die ganzen 12 GB belegt, sondern erst verwendet, wenn diese auch tatsächlich benötigt werden.

#### 4.1.6 Maschine starten und Linux installieren

Nun können wir die Maschine starten und dabei das zuvor heruntergeladene Linux ISO File angeben zum hochstarten. Danach folgen wir einfach den Anweisungen der Installationsroutine.

#### 4.1.7 Linux Virtual Box Treiber

##### **HINWEIS:**

Dies wird nicht benötigt, wenn Sie keine grafische Oberfläche verwenden.

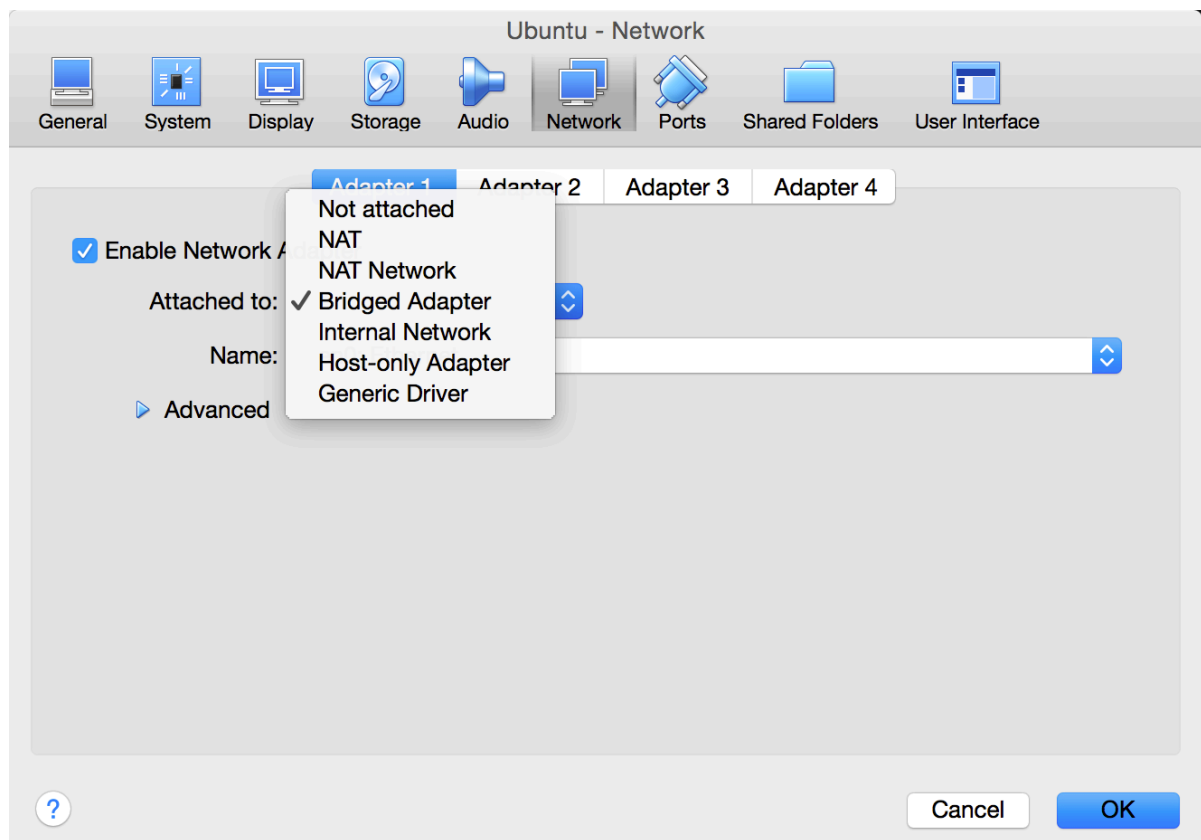
Nachdem wir Ubuntu installiert haben, benötigen wir noch die Treiber für die Desktop Umgebung. Diese können wir ganz einfach Installieren über das Virtual Box selbst unter „Devices -> Insert Guest Additions CD-Image“, oder jedoch über den APT-GET Paket Manager welchen ich in diese fälle empfehlen würde.

Diese können wir ganz einfach mit folgendem Terminal Befehl installieren:

**# sudo apt-get install virtualbox-guest-utils**

Nach der Installation im Anschluss das Ubuntu Linux neustarten, jetzt können wir die Displayauflösung ändern und anpassen.

#### 4.1.8 Netzwerk Übersichtstabelle Zugriffsmöglichkeiten in Virtual Box



Damit wir auf unsere Virtual Box über unser Hauptsystem via Web und SSH zugreifen können, müssen wir in den Virtual Box Einstellungen die Netzwerkkonfiguration auf „Bridged Adapter“ stellen. Eine Übersicht über die Zugriffsmöglichkeiten der Netzwerkeinstellungen sehen sie anhand der Tabelle. Beim Rootserver bei unserem Provider ist dies nicht notwendig, da wir hier von haus aus eine offizielle IP Adresse erhalten haben.

Netzwerktyp	Zugriff Gast -> andere Gäste	Zugriff Host -> Gast	Zugriff Gast -> externes Netzwerk
Not attached	-	-	-

Network Address Translation (NAT)	-	-	✓
NAT Network	✓	-	✓
Bridged Adapter	✓	✓	✓
Internal networking	✓	-	-
Host-only Adapter	✓	✓	-

#### 4.1.9 SSH Server auf dem Virtuellen Server installieren

Nachdem wir das Netzwerk auf „Bridged Adapter“ gestellt haben, bekommt die Virtuelle Linux Maschine eine neue IP-Adresse. Auf diese können wir uns jetzt von unserem Hauptsystem aus verbinden.

Da wir uns mit SSH auf diese Linux Maschine verbinden können, müssen wir zuerst SSH installieren bzw. den dazugehörigen Server.

Öffnen Sie das Terminal und geben im Terminal folgenden Befehl ein:

**# sudo apt-get install openssh-server**

Danach Bestätigen Sie mit Y um die Installation fortzuführen und den Server zu installieren. Nun sollte der Server automatisch gestartet werden und eventuell werden Sie nochmals wegen Ihrem Passwort gefragt.

Nun können wir fortfahren und mit dem nächsten Schritt beginnen.

#### 4.1.10 Mit dem Virtuellen Server vom Hauptsystem per SSH Verbinden

Unter Windows nutzen Sie bitte hierfür ein SSH Tool wie „Putty“

Download von Putty finden Sie hier:

<http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe>

Unter OS X und Linux benötigen Sie keine weiteren Tools, hierfür reicht es das Terminal zu öffnen. Klicken Sie unter OS X auf Spotlight und suchen Sie nach „Terminal“, danach öffnen Sie das Terminal Programm.

Sie erhalten von DigitalOcean nach Fertigstellung eine E-Mail mit den Zugangsdaten zum Server. Hier sehen Sie dann auch die IP-Adresse und das Passwort, welches wir für den Login benötigen.

Wir geben nun in Putty die IP Adresse ein und klicken auf Öffnen und schon werden wir nach dem Login und Passwort gefragt. Hier benutzen wir als Loginnamen „root“, ohne die Anführungszeichen. Unter OS X verwenden wir den Befehl ssh, also geben wir im Terminal als Beispiel folgendes ein:

**# ssh 192.168.0.1 -l root**

Bitte die hier angegebene IP Adresse durch die von Ihrem Server ersetzen.

\*\*\*BILD \*\*\* PUTTY \*\*\* TERMINAL

## 4.2 Linux Installation bei DigitalOcean (Rootserver)



Wenn Sie gleich in einer echten Liveumgebung arbeiten wollen bzw. auch testen wollen, dann kann ich Ihnen nur DigitalOcean empfehlen. Mit den angeführten Link erhalten Sie außerdem bei neu Registrierung einen 10\$ Gutschein und können somit den kleinsten Server ganze 2 Monate nutzen. DigitalOcean bietet Cloudserver an, welche extrem schnelle Anbindung an das Internet haben, als auch eine große Auswahl an Rechenzentren mit hoher Verfügbarkeit. Ihre Daten werden bei DigitalOcean gegen Hardware Schäden automatisch in einem Raid System gesichert. Bedenken Sie jedoch das ein Raid System niemals ein Backup ersetzt!

Hier können Sie Ihren 10 \$ Gutschein einlösen: <https://m.do.co/c/5342cc8df67e>

<b>\$5/mo</b> \$0.007/hour	<b>\$10/mo</b> \$0.015/hour	<b>\$20/mo</b> \$0.030/hour	<b>\$40/mo</b> \$0.060/hour	<b>\$80/mo</b> \$0.119/hour
512 MB / 1 CPU 20 GB SSD Disk 1000 GB Transfer	1 GB / 1 CPU 30 GB SSD Disk 2 TB Transfer	2 GB / 2 CPUs 40 GB SSD Disk 3 TB Transfer	4 GB / 2 CPUs 60 GB SSD Disk 4 TB Transfer	8 GB / 4 CPUs 80 GB SSD Disk 5 TB Transfer

Bild 1: DigitalOcean \$5 Server mit \$10 Gutschein

### 4.2.1 Server „Droplet“ erstellen bei DigitalOcean

Mit wenigen Klicks lassen sich vorgefertigte Linux Distributionen installieren, welche binnen weniger Minuten bereitstehen. Ab der Erstellung eines Servers wird der Server Stundenweise abgerechnet. Das bedeutet, wenn Sie den Server wieder löschen entstehen keine weiteren Kosten mehr. Nach dem Login bei DigitalOcean haben Sie die Möglichkeit ein neues Droplet zu erstellen.

Konfigurieren Sie jetzt einfach Ihre Wunschmaschine, in unserem fall eine Ubuntu 15.10 oder 16.04 x64, sie können auch ein anderes Linux verwenden, jedoch könnte es dann zu Problemen bei den Aufgaben in diesem E-Book kommen bzw. sind die Befehle leicht anders.

Danach geht es weiter mit der Größe, hierfür reicht es für den Anfang zum testen mit dem kleinsten zu beginnen um Kosten zu sparen (0.007 \$ / Stunde).

Geben Sie dann den gewünschten Serverort ein, vorzugweise in Ihrer Nähe, damit die Verbindung zu Ihnen so schnell wie möglich ist.

Zum Schluss geben Sie noch einen Hostnamen für den Server an.

[Droplets](#)[Images](#)[Networking](#)[API](#)[Support](#)

## Create Droplets

Choose an image ?

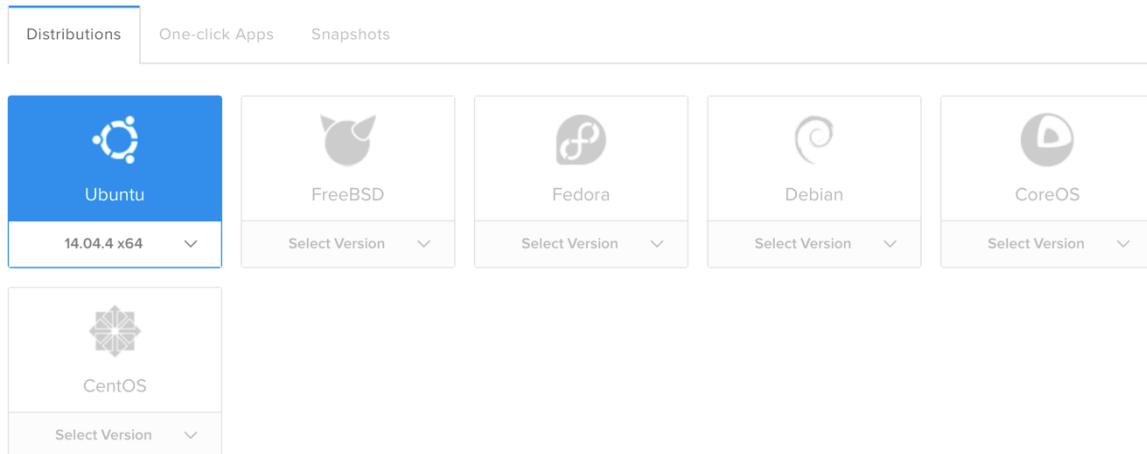


Bild 2: DigitalOcean Server erstellen

### 4.2.2 Mit dem DigitalOcean via SSH Server Verbinden

In der Regel werden auf Server keine Grafischen Oberflächen eingesetzt, wir können von nun an alles per Terminal erledigen und einfach auf den Server Verbinden.

Unter Windows nutzen Sie bitte hierfür ein SSH Tool wie „Putty“

Download von Putty finden Sie hier:

<http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe>

Unter OS X und Linux benötigen sie keine weiteren Tools, hierfür reicht es das Terminal zu öffnen. Klicken Sie unter OS X auf Spotlight und suchen Sie nach „Terminal“, danach öffnen Sie das Terminal Programm.

Sie erhalten von DigitalOcean nach Fertigstellung eine E-Mail (siehe Bild) mit den Zugangsdaten zum Server. Hier sehen Sie dann auch die IP-Adresse und das Passwort, welches wir für den Login benötigen.

Your new Droplet is all set to go! You can access it using the following credentials:

Droplet Name: TBtest  
IP Address: 45.55.82.170  
Username: root  
Password: 041c1ba4223f774f

For security reasons, you will be required to change this Droplet's root password when you login. You should choose a strong password that will be easy for you to remember, but hard for a computer to guess. You might try creating an alpha-numerical phrase from a memorable sentence (e.g. "I won my first spelling bee at age 7," might become "Iwm#1sbaa7"). Random strings of common words, such as "Mousetrap Sandwich Hospital Anecdote," tend to work well, too.

As an added security measure, we also strongly recommend adding an SSH key to your account. You can do that here: [https://cloud.digitalocean.com/ssh\\_keys](https://cloud.digitalocean.com/ssh_keys)

Once added, you can select your SSH key and use it when creating future Droplets. This eliminates the need for root passwords altogether, and makes your Droplets much less vulnerable to attack.

Happy Coding,  
Team DigitalOcean

Bild 3: E-Mail Digital Ocean

Wir geben nun in Putty die IP Adresse ein und klicken auf Öffnen und schon werden wir nach dem Login und Passwort gefragt. Hier benutzen wir als Loginnamen „root“, ohne die Anführungszeichen. Unter OS X verwenden wir den Befehl ssh, also geben wir im Terminal als Beispiel folgendes ein:

**# ssh 192.168.0.1 -l root**

Bitte die hier angegebene IP Adresse durch die von Ihrem Server ersetzen.

Nach Eingabe des Passwortes sollten sie in etwas Folgendes sehen:

**# benutzername@hostname: ~\$**

## 5. Die Linux Grundlagen

### 5.1.1 Verzeichnisstruktur von Linux

Dieser Absatz beschreibt die Verzeichnisstruktur eines Linux-Systems.

In verschiedenen Systemverzeichnissen liegen alle grundlegenden Programme und Konfigurationsdateien. In der Regel können die meisten Verzeichnisse (Ordner) nur mit Root-Rechten (Administrations-Rechten) eingesehen und verändert werden. Dies dient vor allem der Sicherheit des Linux-Systems, womit gerade Anfänger öfters Probleme haben.

In der Regel ist die hier angeführte Verzeichnisstruktur bei den meisten Linux Distributionen gleich bzw. ähnlich. Das Verzeichnis was man unter Windows als „C:“ (Hauptverzeichnis) kennt, ist bei Linux das Root-Verzeichnis „/“. Nachfolgend finden Sie eine Liste zu den gängigsten Verzeichnissen von Linux-Systemen. Eine ausführlichere Information über die Verzeichnisstruktur können Sie in den „Manuals“ nachlesen. Hierzu geben Sie einfach in einem Terminal Fenster den folgenden Befehl ein:

**# man hier**

/	Oberste Ebene, auch „Root“ genannt; Hier sind alle Ordner und Dateien des Systems über Unterordner und Links zu Dateien erreichbar.
---	---

/bin	„binaries“ (Programme); Enthält für Linux unverzichtbare Programme; Diese können im Vergleich zu /sbin von allen Benutzern ausgeführt werden; /bin darf keine Unterverzeichnisse enthalten
/boot	Muss bei Systemstart vorhanden sein; Enthält zum Booten benötigte Dateien; Enthalten sind z.B. Teile des Linux Kernel und Programme wie Memorytest und die Bootloader wie GRUB.
/dev	„devices“ (Geräte); Muss bei Systemstart vorhanden sein; Enthält alle Gerätedateien / Informationen die für den Hardware Betrieb benötigt werden; Gerätedateien für Hot-Plug-Systeme wie USB werden von udev erstellt.
/etc	„et cetera“ (alles Übrige) oder auch „editable text configuration“ (änderbare Text Konfiguration); Muss bei Systemstart vorhanden sein; enthält Konfigurations- und Informationsdateien des Basissystems; Viele wichtige System Konfigurationsdateien für den Betrieb von Server Dienste und Berechtigungen sind hier enthalten. Einige von den Konfigurationsdateien werden überlagert von Konfigurationen aus dem eigenen Homeverzeichnis
/home	„home“ (Heimverzeichnis); Enthält pro Benutzer ein Unterverzeichnis; Jedes Verzeichnis wird nach den Anmeldenamen benannt.
/lib	„libraries“ (Bibliotheken); Muss bei Systemstart vorhanden sein; Die libraries enthalten unverzichtbare Bibliotheken die für den Betrieb von Linux notwendig sind. Unter anderem Kernelmodule, Erweiterungen und Bibliotheken für Programme.
/lost+found	„lost and found“ (Verloren und gefunden); Hier finden Sie Dateien vor die bei defekten Dateien oder Dateisystemen versucht wurden zu Reparieren.
/media	Hier befinden sich Ordner für angeschlossene Speichermedien wie z.B. USB-Sticks, CD-ROM, DVDs.
/mnt	„mount“ (eingehängt); Hier können temporär Festplatten und andere Speichermedien eingehängt werden. Ähnlich wie /media, hier wird jedoch kein Link auf einer Grafischen Oberfläche angelegt.
/opt	„optional“; Ist für die manuelle Installation von Programmen gedacht, welche Ihre eigene Bibliothek mitbringen. Der Speicherort für Selbsterstellte Programme ist im Unterordner /usr/local zu finden.
/proc	„processes“ (Laufende Programme); Muss bei Systemstart vorhanden sein; enthält Schnittstellen zum aktuellen geladenen Kernel und seinen Prozeduren. Dateien hier lassen sich mittels „cat“ auslesen. Beispiele: version (Kernelversion), cpuinfo (Prozessor Information) u.v.m.; Jedes Programm mit Prozessnummern haben einen eigenen Verzeichnisnamen, aus denen man den Status auslesen kann.
/root	Ist das Homeverzeichnis des Superusers „root“. Der Grund, wieso dich das „/root“ Verzeichnis nicht im „/home“ Verzeichnis befindet, liegt darin das Root immer erreichbar sein muss selbst, wenn die Home-Partition aus irgendeinen Grund (Rettungs-Modus, Wartungsarbeiten), nicht verfügbar ist.

/run	Dateien die den Zustand des Gesamtsystems speichern; Inhalt wird bei jedem Neustart der Linux Maschine gelöscht und neu angelegt.
/sbin	„system binaries“ (Systemprogramme); Muss bei Systemstart vorhanden sein; Enthält wichtige Programme für die grundlegenden Aufgaben der Systemverwaltung. Programme in diesem Ordner können nur von Benutzer verwendet werden welche „Superuser“ Rechte haben. Beispiel: „shutdown“ (Herunterfahren), fdisk (Festplatten Partitionieren), mkfs (Formatieren)
/srv	„services“ (Dienste); In der Regel ist dieser Ordner leer, geplant sind jedoch das hier in Zukunft Daten der Dienste liegen.
/sys	„system“; Dateien in diesem Ordner bestehen hauptsächlich aus Kernelschnittstellen.
/tmp	„temporary“ (temporär); Enthält temporäre Dateien und Ordner von Programmen; Verzeichnis wird bei Neustart in der Regel geleert.
/usr	„user“ (Benutzer); Enthält meistens Systemtools, Bibliotheken von installieren Programmen, welche Administratoren oder Benutzer installiert haben welche nicht von der Distribution direkt stammen, und nicht für den Betrieb vom Linux Grundsystem notwendig sind..
/var	„variable“ (Variabel); Enthält Verzeichnisse und Dateien welche von Programmen je nach Bedarf geändert werden. Unter anderem Log Dateien und Aufzeichnungen über Fehlermeldungen; „/var/www“ ist auch meist der Standard Ordner für Web Dateien für den Webserver.

Die Verzeichnisse lassen sich natürlich auch auf unterschiedliche Festplatten und Partitionen anlegen. Dies macht gerade dann sinn, wenn z.b. der „/var/www“ Ordner für alle Webinhalte auf einer eigenen Festplatte zur Verfügung sein soll, genauso wie der Datenbank Ordner. In der Regel und bei kleineren firmen ist jedoch alles auf einer Partition.

## 6. Die Linux Shell

### 6.1.1 Die Bash (Shell)

Die „Bash“ (für Bourne-again shell) ist ein Kommandozeilen-Interpreter. Die Bash wird verwendet um befehle in einem Terminal oder Script auszuführen. In der Regel wenn man in der Kommandozeile auf Linux arbeitet verwendet man eine Bash Shell um Befehle auszuführen. Es gibt natürlich neben der Bash auch noch andere Interpreter welche auch diese Aufgabe übernehmen können, in der Regel werden sie aber mit der normalen Bash arbeiten, sofern sie jetzt nicht spezialisierte Programme selbstschreiben. Mit der Bash lassen sich auch hervorragend sogenannte Bash-Skripte erstellen. Sie erstellen einfach eine Text Datei mit den Befehlen die ausgeführt werden sollen und das Bash-Script arbeitet diese der reihe nach ab.

### 6.1.2 Der Tab-Stopp Trick

Die Bash erleichtert Ihnen auch die Eingabe von Befehlen. Wenn sie nicht den ganzen Befehl aufwendig wissen, reicht es oft nur die ersten Buchstaben anzugeben und anschließend ein bis zweimal auf den Tabulator zu drücken und die Optionen welche Befehle sie nutzen können werden angezeigt bzw. automatisch vervollständigt.

Schreiben sie einmal in die Bash:

```
# cd /usr/lo      + Tabulator Taste, und es wird automatisch auf „local“ vervollständigt.
```

Dieser Trick ist überall in der Bash anwendbar, man kann damit nichts falsch machen.

### 6.1.3 Wildcards / Platzhalter ( \*, ?, [], [a-z], [0-9] )

Eine Wildcard ist ein Symbol oder eine Reihe von Symbolen, welche andere Zeichen vertreten. Sie können zum Beispiel ein Wildcard Symbol verwenden, um eine Liste aller Dateien in einem Verzeichnis anzuzeigen, die mit dem Buchstaben „S“ beginnen.

Die häufigste Wildcard arten in Linux sind das „?“, „\*“ und die eckigen Klammern [].

**Das Fragezeichen „?“**, entspricht ein einzelnes Zeichen zu vertreten. Angenommen wir haben einen Ordner mit vielen Dateien die mit S beginnen und mit „n“ enden, können wir mit den Fragezeichen die exakte Dateilänge definieren.

Zum Beispiel:

**Sonnen** kann auch **S????n** bedeuten, also ersetzt es genau 4 Buchstaben, und es wird eine Liste angezeigt welche genau diese länge von insgesamt 6 Buchstaben haben und die 4 mittleren können frei wählbar sein.

**Der Stern „\*“**, steht für eine oder mehrere Zeichen inklusive Leerzeichen. Anhand der vorigen Beispiels mit **Sonnen** und den 4 Fragezeichen, können wir nun auch **S\*n** benutzen und somit wird der Stern dadurch ersetzt mit allem was dazwischensteht. Hier ist nun ganz egal wie viele Buchstaben dazwischen sind, es wird alles vertreten egal wie lange der Name ist, solange es mit einem „S“ beginnt und einem „n“ endet.

**Die eckige Klammern „[]“**, entsprechen einem definierten Wert. Hier können wir gezielt sagen, das wir alles angezeigt bekommen was einen eigenen Bereich hat. Als Beispiel mit sonnen können wir nun sagen der Bereich umfasst gezielt die Buchstaben „n“o“e“. Dies geschieht ganz einfach Sonnen kann auch bedeuten **S[noe]n**. Damit werden alle Dateien angezeigt welche die gesuchten Buchstaben gezielt beinhalten. Dies funktioniert auch mit ganzen Bereich wie zum Beispiel a-z oder A-Z und nummern 0-9. Als Beispiel **S[a-z]n**.

Wir werden die Wildcard in unserem Tutorial des Öfteren anwenden, gerade der Stern gehört zu den häufigsten Wildcard Symbolen.

## 7. Arbeiten mit Linux

### 7.1 Linux Basis Befehle

Linux befehle und Programme welche Sie ausführen sind in der Regel alle kleingeschrieben bzw. ist auf eine Groß-/Kleinschreibung zu achten. Wir gehen hier ein paar Grundlegende befehle durch mit denen Sie den Umgang mit Linux näher kennenlernen werden.

Hier lernen sie einige Basis Befehle samt den wichtigsten Optionen, wenn ich alle Optionen hier anführen würde, wäre dieses Buch 3-4 x so umfangreich, die meisten Optionen werden sie niemals oder sehr selten benötigen.

### 7.1.1 Sudo und der Superuser (etwas Root User ausführen)

„sudo“ steht für (switch user and do), als wechsele Benutzer und führe aus. Für viele Befehle unter Linux benötigen sie sogenannte Root-Berechtigung, welche sie nur mit diesem Benutzer anwenden können. Damit man nicht permanent mit dem Root Benutzer arbeiten muss gibt es die „sudo“ Funktion. Hier können wir einfach einen Befehl ausführen mit vorangegangenen „sudo“ um den Befehl als „Root Benutzer“ auszuführen.

#### # sudo LinuxBefehl

Alternativ können wir auch mit dem Befehl:

#### # sudo -i

in einen dauerhaften Root Benutzer wechseln. Dies ist vor allem dann sinnvoll, wenn man viele Einstellungen gerade vornehmen und nicht jedes Mal „sudo“ schreiben möchte.

### 7.1.2 Inhalt eines Verzeichnisses ansehen mit „ls“

„ls“ steht für „list directory contents“ (Zeige Ordnerinhalt an). Es ist nicht möglich LS oder Ls zu schreiben, wie oben angeführt ist bei allen befehlen die Groß-/Kleinschreibung wichtig.

„ls“ bietet auch noch eine fülle an zusätzlichen Optionen welche wir benutzen können. Die gebräuchlichsten Optionen sind „-a“ (all) , „-l“ (Lange Listen Format), „-t“ (Sortieren bei Time/Zeit).

Führen Sie folgende befehle aus für die Auflistung des aktuellen Verzeichnisses.

Zeigt den Inhalt des aktuellen Verzeichnisses (Versteckte Dateien werden ausgeblendet):

# ls

Zeigt den Inhalt des aktuellen Verzeichnisses (Versteckte Dateien werden angezeigt):

# ls -a

Zeigt den kompletten Inhalt des aktuellen Verzeichnisses mit Langen Listen Format:

# ls -la

Zeigt den kompletten Inhalt des aktuellen Verzeichnisses mit Langen Listen Format sortiert nach Datum:

# ls -lat

Zeigt den kompletten Inhalt der obersten Root Ebene in einer Langen Listen Format:

# ls -la /

Zeigt den kompletten Inhalt des „/bin“ Verzeichnisses in einer Langen Listen Format:

# ls -la /bin

Zeigt alle Optionen und Möglichkeiten von „ls“ an:

**# ls --help**

Zeigt Ihnen alle Dateien und Ordner an die mit „s“ beginnen, der Stern ersetzt sozusagen alle fehlenden Zeichen.

**# ls s\***

### 7.1.3 Verzeichnis wechseln mit „cd“

„cd“ steht für „change directory“ (Ordner wechseln). Und das ist es auch was dieser Befehl tut, er wechselt den aktuellen Ordner in dem Sie sich gerade befinden. Hier zu geben sie einfach den Befehl „cd“ und Ordnernamen an in den Sie wechseln möchten und schon sind sie innerhalb dieses Ordners. Im Anhang finden sie jetzt einige Beispiele für die Anwendung von „cd“.

Wechseln in den „/bin“ Ordner:

**# cd /bin**

Wechseln in den „/usr/local“ Ordner:

**# cd /usr/local**

Wechseln in den Benutzerordner des aktiven Benutzers:

**# cd**

Mit einem „.“ (Punkt) ist das jetzige Verzeichnis gemeint in das sie sich gerade befinden

**# cd .**

Mit zwei „..“ (Punkten) und der Eingabe, springen Sie eine Verzeichnisebene höher. Sie befinden sich zum Beispiel in „/home/benutzername“, nach dem Befehl sind sie nur mehr im Ordner „/home“.

**# cd ..**

Führt den letzten „cd“ Befehl nochmals aus.

**# cd -**

Um in Ihren Benutzerordner zu wechseln reicht ein einfaches „cd“ und das System weiß das sie in Ihren Ordner zurückkehren wollen.

### 7.1.4 Wo befinde ich mich gerade mit „pwd“

„pwd“ steht für „print name of working directory“. In der Regel bedeutet dies nichts anderes als es Ihnen den aktuellen Ordner anzeigt in dem sie sich gerade aufhalten.

Kombinieren sie „pwd“ mit den „cd“ Befehlen vom vorherigen Kapitel und sehen sie was passiert.

Zeigt den Ordnernamen an in dem sie sich gerade befinden:

**# pwd**

### 7.1.5 Zeige mir den Inhalt einer Datei mit „cat“

„cat“ steht für „concatenate files and print on the standard output“. Mit „cat“ können sie den Inhalt von Dateien direkt im Terminal ausgeben lassen. Zudem bietet „cat“ wieder eine menge an zusätzlichen Optionen an, um zum Beispiel bei großen Dateien eine Zeilen Limitierung anzugeben damit nicht die ganze Datei angezeigt werden muss.

Zeigt den Inhalt von „cpuinfo“ an:

```
# cat /proc/cpuinfo
```

Zeigt die Zeilennummer vom Inhalt der Datei „cpuinfo“ an:

```
# cat -n /proc/cpuinfo
```

Zeigt alle weiteren Informationen zu „cat“ an:

```
# man cat
```

### 7.1.6 Anzeige von Werten mit „echo“

„echo“ zeigt Ihnen Texte, Werte von Variablen an welche in Linux vordefiniert sind oder welche sie auch noch definieren können. Variablen sind üblicherweise in der Linux Bash (Terminal) in Großbuchstaben.

Geben Sie folgenden Befehl ein um ins Terminal einen Auszug aus Ihrem Text zu machen:

```
# echo Dies ist ein Test Text
```

Zeigt ihr „/home“ Verzeichnis an:

```
# echo $HOME
```

Diese Variable zeigt Ihnen an in welchem Verzeichnis sie zuvor waren:

```
# echo $OLDPWD
```

Zeigt an welche Pfade definiert worden sind:

```
# echo $PATH
```

Es werden unter anderem werden dann automatisch in den angeführten pfaden Befehle zur Verfügung gestellt wie zum Beispiel die ganzen Befehle welche wir hier durchgehen. Findet Linux einen gewünschten Befehl im ersten Pfad z.B. „/usr/local/bin“, so sucht dieser nicht weiter im nächsten Pfad „/bin“.

Schauen wir uns einmal an in welchen Ordner der Befehl cat liegt:

```
# which cat
```

Und nun das selbe mit tac:

```
# which tac
```

„tac“ ist das selbe wie „cat“ nur rückwärts geschrieben. Raten sie einmal was dieser Befehl nun macht..... Genau, er macht das selbe wie „cat“ nur die Ausgabe ist nun rückwärts!

Durch „which“ sehen wir nun wo das Programm liegt welches wir ausgeführt haben, und die \$PATH variable hat in diesen Ordnern automatisch gesucht.

Weitere Linux Variablen zur Bash finden sie unter diesem Link im Bash-Scripting Guide:  
<http://www.tldp.org/LDP/abs/html/internalvariables.html>

#### 7.1.7 Dokumentationen zu Befehlen ansehen mit „man“

„man“ steht für „manuals“ (Handbücher). Zu fast jedem Befehl unter Linux gibt es ein Handbuch. Diese lässt sich ganz einfach wie wir schon gelernt haben aufrufen mit „man BefehlsName“. Sie können aber auch gezielt nach Handbüchern suchen wo sie ungefähr einen Namen wissen, hierfür nutzen wir die Option „-k“.

Zeigt alle verfügbaren Handbücher an in denen das Wort „zip“ vorkommt.

```
# man -k zip
```

Zeigt das Handbuch zu dem Befehl „ls“ an

```
# man ls
```

Mit „Enter“ oder „Pfeil nach unten“ können sie innerhalb des Handbuches hinunter blättern, mit „Bild runter / Bild raus“ können sie seitenweise blättern, mit „q“ können Sie das Handbuch wieder verlassen und mit „h“ innerhalb des Handbuches erhalten Sie weitere Hilfe zum Umgang innerhalb von „man“.

#### 7.1.8 Verlassen einer aktuellen Sitzung mit „exit“

Mit „exit“ beenden sie die laufende Sitzung ihres Terminal-Fenster. Sind sie via SSH auf einen Server verbunden, so wird die Verbindung zum Server getrennt.

#### 7.1.9 Den Bildschirminhalt löschen mit „clear“

Mit „clear“ können sie Ihr Terminal-Fenster löschen. Dies ist gerade dann sinnvoll, wenn wichtige Informationen nicht mehr sichtbar am Bildschirm angezeigt werden wollen.

**Hinweis:** Sie können jedoch noch mit der „Bild auf“ taste zum Verlauf gelangen.

#### 7.1.10 Die „help“ Option

Viele Programme haben zusätzlich zum Manual auch eine eingebaute Hilfe. Versuchen sie einfach den Befehl auszuführen mit zwei Bindestrichen gefolgt von help.

Beispiel:

```
# ls --help
```

```
# gzip -h
```

Sollte „--help“ nicht funktionieren können sie es noch mit „-h“ stattdessen probieren, falls dies auch nicht klappt bleibt Ihnen nur mehr das „man“ übrig, hilft dies auch nicht gibt es keine weiteren Informationen / Optionen zu dem Programm.

## 7.2 Arbeiten mit Dateien und Ordner

### 7.2.1 Erstellen und Löschen eines neuen Ordners (mkdir, rmdir, rf)

„**mkdir** ordnername“ (make directory) Erstellen eines leeren Ordner.

Mit der Option „-p“ können sie auch gleich mehrere Unterordner erstellen.

Erstellt einen Ordner namens „testordner“ im aktuellen Verzeichnis:

```
# mkdir testordner
```

Versucht zwei Ordner zu erstellen, der erste Befehl schlägt jedoch fehl da die Option „-p“ (parent) nicht angegeben wurde.

```
# mkdir testordner2/testordner
```

```
# mkdir -p testordner2/testordner
```

„**rmdir** ordnername“ (remove directory) Löschen eines leeren Ordner.

Mit der Option „-p“ können sie auch gleich mehrere Unterordner löschen.

Löscht den leeren „testordner“:

```
# rmdir testordner
```

Löscht beide Ordner sofern diese Leer sind, da die Option „-p“ angegeben wurde:

```
# rmdir -p testordner2/testordner
```

„**rm -rf** ordnername“ (Remove) Löschen eines Ordners mit dem gesamten Inhalt.

Mit dem „rm -rf“ befehl und den Ordnernamen können sie komplette Ordner samt den gesamten Inhalten löschen. Seien sie Vorsichtig mit dieser Funktion das sie nicht unabsichtlich zu viel löschen!

### 7.2.2 Erstellen und Löschen einer neuen Datei

„touch dateiname“ Erstellen einer Datei ohne Inhalt.

```
# touch meinedatei
```

„rm dateiname“ Löschen einer Datei.

```
# rm meinedatei
```

Mit der Option „-r“ können wir Dateien und Ordner rekursiv löschen, die Option „-f“ steht für „Force removal“ und verhindert das sie nochmals jeden Löschvorgang extra bestätigen müssen.

```
# rm -r Ordnername
```

```
# rm -f Dateiname
```

Wenn Sie alle Dateien in einem Ordner löschen wollen welche mit einem „s“ beginnen, so können sie ein Stern hintendran setzen.

```
# rm -f s*
```

### 7.2.3 Kopieren von Dateien und Ordnern

„cp“ steht für „copy“ Damit lassen sich Dateien und Ordner Kopieren

Geben Sie einfach „cp dateinamen neuerdateiname“ an.

Kopiert die Datei in den selben Ordner:

```
# cp datei1 datei2
```

Kopiert die Datei gleich in einen neuen Ordner im Beispiel „/tmp“:

**# cp datei1 /tmp/datei2inTMPordner**

Kopieren eines Ordners mit der Option „-r“ (rekursiv):

**# cp -r Ordner1 Ordner2**

#### 7.2.4 Bewegen und Umbenennen von Dateien und Ordnern

„mv“ steht für „move“ Damit können sie Dateien und Ordner verschieben od. umbenennen

Wenn sie einen Ordner oder eine Datei in den selben Ordner verschieben in dem sich der Ordner bzw. Datei gerade befindet, dann wird diese lediglich umbenannt.

Dieser Befehl benennt Ordner1 in Ordner2 um:

**# mv Ordner1 Ordner2**

Diese Befehl verschiebt Ordner1 in das Verzeichnis „/tmp“

**# mv Ordner1 /tmp**

#### 7.2.5 Dateien Inhalte anzeigen (cat, more, less, head, tail, sort)

1. „cat dateiname“ Zeigt den Inhalt einer Datei an.
2. „more dateiname“ Zeigt den Inhalt schrittweise einer Datei an.
  - a. Enter für die nächste Zeile
  - b. Leertase für die nächste Seite
3. „less dateiname“ Wie „more“ jedoch mit mehr Funktionen
4. „head dateiname“ Zeigt die ersten 10 Zeilen einer Datei an.
5. „tail dateiname“ Zeigt die letzten 10 Zeilen einer Datei an.
  - a. „tail -20 dateiname“ Zeigt die letzten 20 Zeilen an.
  - b. „tail -f dateiname“ Zeigt fortlaufend Veränderungen an, sobald etwas neues in die datei geschrieben wurde. (z.B. in Log Dateien)
6. „sort dateiname“ Zeigt den Inhalt einer Datei an und sortiert diese in einer Alphabetischen reihenfolge.
  - a. „sort -u dateiname“ Können sie Duplikate ausblenden
  - b. „sort -r dateiname“ Zeigt den Inhalt verkehrtherum an

#### 7.2.6 Nano Editor , eine Datei bearbeiten

Nano ist nur einer der unzähligen Editoren für die Kommandozeile.

Der Editor bietet jedoch einige Vorteile gegenüber anderen Editoren:

1. Nano ist ein einfacher Editor
2. Leicht zu erlernen
3. Nicht so umfangreich und kompliziert wie „vi“ oder „emacs“
4. Wenn „nano“ nicht verfügbar ist, versuchen sie es mit „pico“

Sie können Nano starten mit dem Befehl „nano dateiname“.

**# nano dateiname**

Im Anschluss sehen sie eine menge Funktionen an der Unterseite von nano. Mit dem angezeigten „^“ ist die CTRL (STRG) Taste gemeint.

Die wichtigsten Funktionen sind:

**STRG + W = Suchen**

**STRG + O = Speichern**

**STRG + X = Datei verlassen**

**STRG + G = Hilfe**

Der Editor kann noch eine menge mehr an Funktionen, diese sind jedoch für uns vorerst die wichtigsten, da man diese im täglichen gebrauch oft benötigt.

### 7.2.7 Eine Datei ausführen

Um Programme auszuführen die nicht im \$PATH stehen gibt es die Möglichkeit diese direkt aufzurufen.

Beispiel Sie haben ein Programm namens „ausführbareDatei“ im Ordner „/tmp“.

Nun können Sie diese folgend starten:

Direkt über den vollen Pfad mit  
# /tmp/ausführbaredatei

Oder Sie sind im selben Ordner und verwenden nur einen „./“ mit Dateiname.  
# cd /tmp  
# ./ausführbaredatei

### 7.2.8 Archivieren von Dateien und Ordnern mit „tar“

Mit „tar“ können wir mehrere Dateien und Ordner in eine einzige Datei verpacken und diese für die Archivierung oder Versand vorbereiten. Bei tar benötigen wir einige Optionen, die wichtigsten hiervon sind:

1. „-c“ (create) Erstellen eines tar Archives.
2. „-x“ (extract) Entpacken der Dateien eines tar Archives.
3. „-t“ (table) Zeigt den Inhalt eines tar Archives an.
4. „-v“ (verbose) Zeigt an woran tar gerade arbeitet.
5. „-z“ (compression/zip) Benutzt eine Kompression
6. „-f“ (file) Gibt an das es diese Datei benutzen soll

Erstellen wir ein tar Archiv mit dem Namen Report12.tar der den Inhalt vom ganzen „/tmp“ Ordner beinhaltet.

**# tar cvf Report12.tar /tmp**

Derselbe Befehl jedoch mit Kompression „-z“ und .tgz (Tar Gzip) Endung:

**# tar zcvf Report12.tgz /tmp**

Inhalt des erstellten tar Archives anzeigen:

```
# tar tf Report12.tar
```

Entpacken eines Tar Archives:

```
# tar xvf Report12.tar
```

Derselbe Befehl jedoch mit Kompression „-z“:

```
# tar zxvf Report12.tgz
```

### 7.2.9 Komprimieren von Dateien und Ordnern mit „Zip, gzip“

„gzip dateiname“ Um Dateien oder Ordner zu komprimieren und Speicher zu sparen.

„gunzip dateiname“ Um eine Zip Datei wieder zu entpacken.

### 7.2.10 Dateigrößen Anzeigen mit „du“

„du“ steht für „Disk Usage“ und zeigt die Dateien und Ordner an wieviel Speicher diese auf der Festplatte belegen.

Zeigt die Größe der Datei in Bytes an:

```
# du dateiname
```

Zeigt die Größe aller Dateien im Ordner in Bytes an:

```
# du *
```

Zeigt die Größe aller Dateien im Ordner in Kilobytes an:

```
# du -k *
```

Zeigt die Größe aller Dateien im Ordner in einem verständlichen Format an:

```
# du -h *
```

### 7.2.11 Dateien und Ordner suchen und finden mit „locate“ und „find“

Mit „locate“ können Sie Dateien und Ordner durchsuchen im gesamten Linux System. Hierfür verwendet „locate“ eine eigene Datenbankdatei welche schnell abgerufen wird. Hier kann es natürlich vorkommen das nicht alle Dateien und Ordner vorhanden sind, sofern diese Datenbankdatei noch nicht erneuert (Updated) wurde.

```
# locate dateiname
```

Zusätzlich gibt es noch die Option „-i“ um die Groß-/Kleinschreibung zu ignorieren:

```
# locate -i dateiname
```

Der Vorteil gegenüber find liegt in der Schnelligkeit, denn eine Datenbankdatei zu durchsuchen welche alle Dateinamen und Ordernamen beinhaltet ist wesentlich schneller als die gesamte Festplatte erneut zu durchsuchen.

Um die locate Datenbank zu initialisieren können wir ein Update anstoßen mit:

```
# sudo updatedb
```

Mit „find“ können wir Dateien und Ordner durchsuchen, kombiniert mit dem „sudo“ Befehl werden auch Ordner durchsucht für welche sie normalerweise keine Berechtigungen haben.

Eine klassische „find“ suche in „/root/“ Ordner sieht so aus :

**# find / -name dateiname**

Kombinieren lässt sich das auch noch mit der Type ob es ein Ordner oder eine Datei ist:

**„-type d“ für Ordner (directory)**

**„-type f“ für Datei (file)**

Beispiel:

**# find / -type d -name dateiname**

Find bietet noch eine fülle an weiteren sinnvollen Funktionen wie die suche nach Dateien welche in den letzten 1-2 Tagen bearbeitet wurden, welche die eine bestimmte Größe haben.

Suche nach Dateien welche größer als 10 Megabyte sind:

Größen Typen: c = bytes, k = Kilobytes, M = Megabyte , G = Gigabytes

**# find / -type f -size 10M -name dateiname**

Suche nach Dateien welche in den letzten 48 Stunden bearbeitet wurden:

**# find / -atime 2 -name dateiname**

Suche nach Dateien welche in den letzten 10 Minuten bearbeitet wurde:

**# find / -cmin 10 -name dateiname**

Weitere Informationen zu „locate“ und „find“ gibt es wie üblich in den „man“ Pages.

**# man find**

**# man locate**

## 7.2.12 Filtern von der Ausgabe mit Pipes, Grep, Sort ( | )

Unter einer Pipe in Linux versteht man den langen Strich welcher sich auf der deutschen Windows Tastatur links neben dem „Y“ befindet. Dieser ist mit gedrückter ALT Taste zu erreichen. Unter Mac OS X befindet sich diese Taste versteckt hinter der Nummer „7“ mit gedrückter ALT Taste. So erhalten sie dann ein Pipe: „|“.

Wir können Pipes verwenden um 2 oder mehr Befehle gleichzeitig auszuführen welche wir miteinander kombinieren, und somit komplexere aufgaben durchführen zu können.

Wir können als Beispiel den „cat“ Befehl verwenden in Kombination mit einem „grep“ herausfischen von bestimmten Inhalten.

Legen wir los mit einem „cat“ auf eine Log Datei:

**# cat /var/log/dmesg**

Danach sollten sie einen Auszug auf der Log Datei sehen und können jetzt gezielte Inhalte herausfischen, dies können sie ganz individuell gestalten. In meinem falle nutze ich „grep“ um Inhalte herauszufischen welche das Wort „sda“ beinhalten.

```
# cat /var/log/dmesg | grep sda
```

In umgekehrter Reihenfolge können wir „-i“ anwenden, so wird alles angezeigt außer diesen Zeilen welche das Ziel Wort „sda“ enthalten:

```
# cat /var/log/dmesg | grep -v sda
```

Wenn der Inhalt zulange ist und man nicht Scrollen möchte, kann man diese auch noch mit „more“ oder „pg“ kombinieren

```
# cat /var/log/dmesg | grep -v sda | more
```

```
# cat /var/log/dmesg | grep -v sda | pg
```

Wenn sie wissen wollen wie viele Zeilen, Wörter insgesamt in der Datei „dmesg“ stecken könne Sie diese auch mit dem „wc“ (Word Count) Befehl kombinieren.

```
# cat /var/log/dmesg | wc
```

Die Ausgabe zeigt dann 3 Zahlen an, die erste definieren wie viele Zeilen, die mittlere wie viele Wörter, und die letzte wie viele Buchstaben/Zeichen enthalten sind.

Mit „sort“ können wir die Ausgabe auch noch vor der Anzeige sortieren, es ist auch gut zu verstehen das die Ausgabe von dem ersten Befehl an den angehängten Befehl nach der pipe übergeben wird.

Die 2 folgende befehlen haben die selbe Ausgabe:

```
# sort /var/log/dmesg
```

```
# cat /var/log/dmesg | sort
```

Der Unterschied hierbei ist jedoch das beim zweite Befehl zuerst die Datei ausgegeben wird und im Anschluss sortiert wird, bevor diese auf den Bildschirm erscheint, der erste Befehl führt gleich bei der Datei Ausgabe die Sortierung automatisch durch.

Durch solche Kombinationen und mehreren pipes lassen sich so mächtige kleine Skripte bauen. Denn Sie können auch alle miteinander kombinieren wie zum Beispiel hier gezeigt:

```
# cat /var/log/dmesg | grep -v sda | grep md | sort -r | grep usb
```



**Die Ausgabe wird von Befehl zu Befehl weitergereicht und bearbeitet.**

Weitere Informationen zu grep, sort, wc und andere Befehle wie üblich in den „man“ Pages:

```
# man grep
```

```
# man sort
```

```
# man wc
```

### 7.2.13 Ausgabe in eine neue Datei speichern und in bestehenden anhängen

Nachdem wir nun das wichtigste über pipes gelernt haben wollen wir auch die Ausgabe nutzen um diese in eigene Dateien zu schreiben und zu speichern.

Wenn sie Befehle ausführen die auf dem Bildschirm erscheinen und diese Ausgabe in eine neue Datei speichern wollen gibt es hierfür ganz einfach den Zusatz-Befehl „>“ und „>>“.

Schauen wir uns einmal die Ausgabe von folgenden Befehl an:

```
# echo „Hallo Welt“
```

Jetzt wollen wir diese Ausgabe in eine neue Datei namens „MeineNeueDatei“ im „/tmp/“ Ordner speichern:

```
# echo „Hallo Welt“ > /tmp/MeineNeueDatei
```

Jetzt wird die ganze Ausgabe in diese „MeineNeueDatei“ gespeichert, und bei jedem Aufruf dieses Befehls wird erneut die Datei überschrieben und die alten Inhalte gelöscht.

Wir haben jedoch auch die Möglichkeit nicht die Datei zu überschreiben sondern die Ausgabe am Ende der Datei anzufügen, dies macht gerade dann sinn wenn man Log Dateien beschreibt wo alte Informationen erhalten bleiben sollen. Hierfür nutzen wir einfach die doppelte Ausführung für anhängend „>>“:

```
# echo „Hallo Welt“ >> /tmp/MeineNeueDatei
```

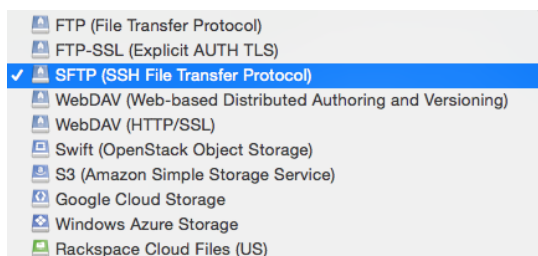
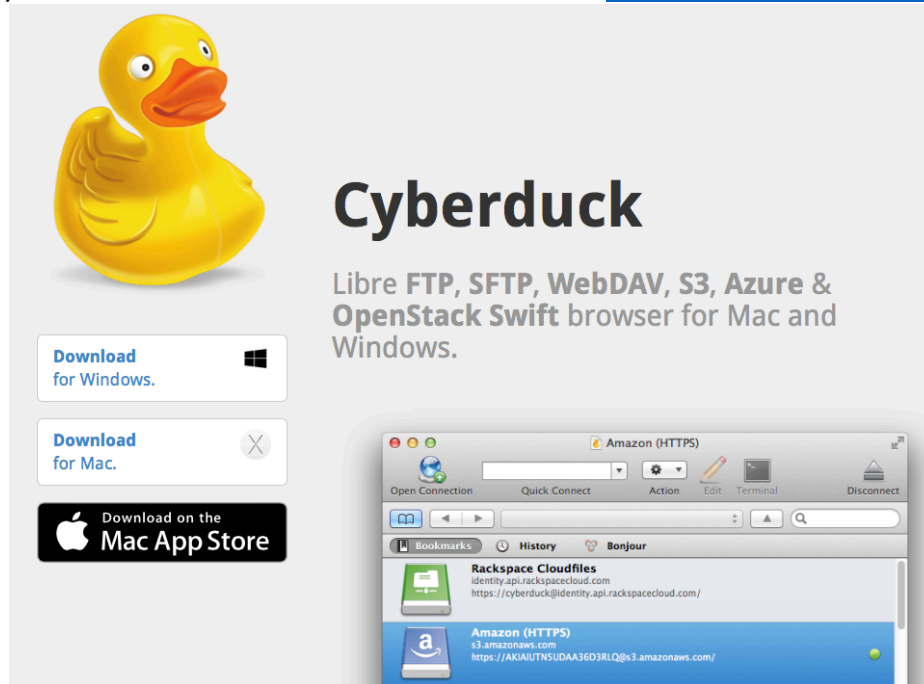
Sie können nun überprüfen ob alles korrekt funktioniert in dem Sie ein „cat“ auf die neue Datei anwenden. Führen sie den vorhergehenden Befehl mit den „>>“ mehrmals aus und sehen Sie was dann hier passiert:

```
# cat /tmp/MeineNeueDatei
```

#### 7.2.14 Dateien sicher und verschlüsselt kopieren über das Netzwerk mit SSH und SFTP

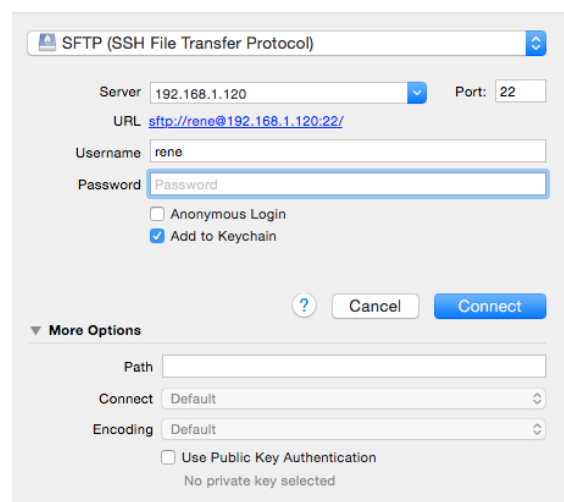
Für die sichere Dateiübertragung zwischen Ihrem PC und dem Linux Rechner nutzen wir ein komfortables FTP Programm. Ich empfehle hierfür das kostenlose Programm „Cyberduck“. Aber Achtung im Mac App Store kostet Cyberduck etwas, da sie sich mit den geldern finanzieren, man kann es jedoch kostenlos direkt über die Herstellerseite beziehen.

Cyberduck Herstellerwebseite finden Sie unter: <https://www.cyberduck.io/>



Nach der Installation und dem Programmstart müssen Sie eine neue Verbindung aufbauen, mit einem klick auf „Open Connection“. Wichtig hierbei ist da wir eine sichere Verbindung wollen das Sie „SFTP (SSH File Transfer Protocol)“ verwenden.

Nach Angabe der Server IP Adresse und den Linux Benutzernamen mit Passwort, steht der sicheren Dateiübertragung nichts mehr im Wege.



## 7.3 Eigentum und Zugriffsrechte

### 7.3.1 Besitzer von Dateien und Ordner mit chown

Jeder Ordner und jede Datei in Linux hat Ihre eigenen Berechtigungen, einen Eigentümer und eine Gruppe. Der Eigentümer kann seine eigenen Ordner und Dateien nur an eine Gruppe ändern zu der er gehört. Dem Superuser (root) ist es möglich die Gruppe aller zu ändern.

Um zu sehen wer der Eigentümer und die Gruppe einer Datei ist können wir den Befehl:

**# ls -l**

verwenden. Die Ausgabe jeder Datei / Ordner sieht dann in etwa so aus:

```
-rw-rw-r-- 1 rene rene 140 Jan 09 10:45 meinedatei.txt
```

Das erste „rene“ im Feld zeigt den Eigentümer der Datei an und das zweite „rene“ zeigt die Gruppe an zu welcher die Datei gehört.

Auch jedes laufende Programm/Prozess bei einem Linux System haben einen Eigentümer. Mit dem Befehl:

**# ps -u**

können wir uns die Prozesslisten ansehen mit dem dazugehörigen Eigentümer.

Um eine Datei einen neuen Eigentümer oder einer neuen Gruppe zuzuordnen gibt es den Befehl „chown“ (change owner).

Der Aufbau des Befehls ist folgender „**chown NeuerBenutzer:NeueGruppe Dateiname**“

Unsere Beispieldatei namens „**meinedatei.txt**“ welche dem Eigentümer und der Gruppe „rene“ gehört, ändern wir nun den Eigentümer auf „alex“ und die Gruppe auf „webserver“. Dazu passen wir unseren Befehl an:

**# chown alex:webserver meinedatei.txt**

Überprüfen können wir dies wieder mit „ls -l“ und haben die Eigentümer-Berechtigungen nun abgeschlossen.

### 7.3.2 Zugriffsrechte auf die Dateien und Ordner

Damit nicht jeder Benutzer auf alle Dateien und Ordner zugriff hat, gibt es für alle eigene Zugriffsrechte. Hier können wir definieren welcher Benutzer, Gruppe, Gast zugriff auf einen Ordner oder Datei hat.

Es gibt drei Arten von Zugriffsrechten:

**Lesen**, der Benutzer darf die Datei lesen oder nicht. Für Ordner gilt das der Inhalt angezeigt werden kann oder nicht.

**Schreiben**, der Benutzer darf die Datei bearbeiten oder im Ordner neue Dateien anlegen.

**Ausführen**, der Benutzer darf die Datei ausführen. Das Ausführrecht für Ordner bedeutet, dass der Benutzer die Datei ausführen darf wenn der Dateiname bekannt ist, jedoch nicht den Ordner auflisten kann.

Daneben gibt es auch noch drei Arten von Benutzern:

**Eigentümer**, bestimmt was man mit der Datei/Ordner tun kann.

**Gruppe**, bestimmt was Mitglieder der Gruppe mit der Datei/Ordner tun können

**Gast und Mitglieder anderer Gruppen**, bestimmt was jeder mit der Datei tun kann.

Der Eigentümer oder Superuser kann festlegen wer mit der Datei was machen kann. Es ist zum Beispiel auch möglich das der Eigentümer mit der Datei alles machen darf, die Gruppe jedoch nur Leserechte hat und der Gast keinen Zugriff darauf hat.

### 7.3.3 Zugriffsrechte unter Linux

Insgesamt gibt es neun Berechtigungszeichen, welche steuern wer den Datei/Ordnerinhalt lesen, schreiben und ausführen darf. Die ersten drei Zeichen (ROT) bestimmen die Berechtigung für den Eigentümer. Die nächsten drei Zeichen (BLAU) bestimmen die Berechtigung für die Mitglieder der Gruppe und die letzten drei Zeichen (GRÜN) bestimmen die Berechtigungen für den Gast und andere Gruppenmitglieder.

Führen wir wieder wie zuvor unser „ls -l“ aus , erhalten wir als Beispiel folgendes:

```
# -rw-rw-r-- 1 rene rene 140 Jan 09 10:45 meinedatei.txt
```

Die Berechtigungen werden mit Zeichen (**rw**x) angezeigt. Im Beispiel „**rw**-“ (Rot) für Eigentümer ist gemeint das dieser Lese (Read) und Schreibrechte (Write) hat. Genauso wie die Gruppe mit „**rw**-“ (Grün). Beim Gast jedoch sieht man nur ein „**r**--“dies bedeutet das dem Gast die Schreibrechte fehlen. Wenn wir eine ausführbare Datei hätten würde hier auch ein x (execute) angezeigt werden, als Beispiel „**rw**x“. In dem Beispiel gibt es keine Ausführrechte.

### 7.3.4 Zugriffsrechte setzen mit chmod

Unter Linux verwendet man für die Zugriffsrechte zum verwalten den Befehl „**chmod**“.

Nur der Eigentümer der Datei und der Superuser können diesen Befehl benutzen.

Anwenden kann man „chmod“ ganz einfach mit: „**chmod Rechte Dateiname**“

Wir können „chmod“ mit zwei unterschiedlichen Modi nutzen, anhand derer wir die Rechte ändern möchten. Den Symbolischen Modus und den Oktal-Modus.

#### „chmod Oktal Modus“

Mit dieser Methode nutzen wir jeweils drei Ziffern für die Rechte des Eigentümer, der Gruppe und des Gastes. Eine 1 steht für das Ausführen, die 2 für das Schreiben und die 4 für das Lesen. Eine 0 bedeutet keine Zugriffsrechte. Um die benötigten Rechte anzugeben

müssen diese Nummern addiert werden. Zum Beispiel ein Lese + Schreibrecht werden 4 und 2 addiert, was einer Berechtigungszahl von  $(4+2) = 6$  entspricht.

Liste aller möglichen Werte:

Berechtigung	Wert
---	0
--x	1
-w-	2
-wx	3
r--	4
r-x	5
rw-	6
rwX	7

Einige Beispiele zum Verständnis mit der Datei „meinedatei.txt“:

Vollzugriff auf Eigentümer ( $4+2+1 = 7$ ), Gruppe ( $4+2+1 = 7$ ) und Gast ( $4+2+1 = 7$ ):

**# chmod 777 minedatei.txt**

Vollzugriff auf Eigentümer, Gruppe ( $4+2+1 = 7$ ). Kein Zugriff (0) für den Gast:

**# chmod 770 minedatei.txt**

Schreib+Lesezugriff für Eigentümer ( $4+2 = 6$ ), Lesezugriff für die Gruppe (4),

Lese+Ausführzugriff ( $4+1 = 5$ ) für Gast:

**# chmod 645 minedatei.txt**

### „chmod Symbolischer Modus“

Wer sich nicht mit Zahlen anfreunden kann, der hat auch die Möglichkeit diese mit Buchstaben zu setzen. Als Buchstaben werden „r“ (read) für das Leserecht, „w“ (write) für das Schreibrecht und der Buchstabe „x“ (eXecute) für das Ausführrecht. Mit einer Kombination dieser Buchstaben wird angegeben, welche Rechte wir setzen oder entfernen. Dazu wird + und - eingesetzt um Rechte hinzuzufügen oder zu entfernen. Mit einem „=“ werden die Rechte explizit angegeben festgesetzt und die restlichen entzogen.

Mit dieser Methode müssen Sie auch angeben für wem Sie die Berechtigungen setzen wollen für den Eigentümer mit „u“ (User), der Gruppe mit „g“ (Group) und dem Gast/Andere mit „o“ (other).

Schauen wir uns wieder „meinedatei.txt“ an:

Ursprünglich hat die Datei folgende Rechte:

**# -rw-rw-rw- 1 rene rene 140 Jan 09 10:45 minedatei.txt**

Mit einem „-“ entfernen wir die Lese (r) und Schreibrechte (w) für den Gast „o“:

**# chmod o-rw minedatei.txt**

Ausgabe:

**#-rw-rw--- 1 rene rene 140 Jan 09 10:45 minedatei.txt**

Nun fügen wir dem Eigentümer (u) und der Gruppe (g) noch Ausführungsrechte hinzu:  
**# chmod ug+x meinedatei.txt**

Ausgabe:

**#-rwxrwx--- 1 rene rene 140 Jan 09 10:45 meinedatei.txt**

Ganz egal wie sie „chmod“ anwenden, die beiden Wege führen zum Ziel. Meistens ist es jedoch einfacher sich mit den Zahlen zu befassen, diese kommen auch üblicherweise in den meisten Dokumentationen vor, wenn es um das Installieren von Programmen oder PHP Dateien wie Wordpress und anderen System geht.

### 7.3.5 Dateitypen erkennen

Im vorhergehenden Beispiel haben wir die Zugriffsrechte gesetzt, im neuen Beispiel haben wir bei einem „ls -l“ folgende Ausgabe:

```
# -rw-rw-r-- 1 rene rene 140 Jan 09 10:45 meinedatei.txt
# drwxrwxr-- 1 rene rene 140 Jan 09 10:45 home
# lrwxrwxrwx 1 rene rene 140 Jan 09 10:45mlink -> meinedatei.txt
```

Nun sehen wir beim ersten Zeichen ein „-“, „d“ und ein „l“. Dieses erste Zeichen gibt den Dateityp an, ob es sich um eine Datei, einen Ordner (d) oder einer Verknüpfung (l) handelt.

Neben den drei angeführten gebräuchlichsten Dateitypen gibt es unter Linux folgende Dateitypen:

1. „-“, reguläre Datei
2. „d“, Ordner (directory)
3. „l“, Verknüpfung / Verweis auf anderen Ordner oder Datei (link)
4. „c“, zeichenorientierte Gerätedatei
5. „b“, blockorientierte Gerätedatei
6. „s“, Unix Domain Sockets
7. „p“, benannte Pipe

## 7.4 Benutzer- und Gruppenverwaltung

### 7.4.1 Der Root-User (Administrator)

Der **Root-User** auch als der **Super User** oder **Administrator** bekannt, ist ein spezielles Benutzerkonto in Linux, das man für die Systemverwaltung verwendet.

Der Root-User ist das Konto, welche auf alle Befehle und Dateien Zugriff hat. Der Root-User kann viele Aktionen ausführen, die ein normaler User nicht kann, wie zum Beispiel die Installation neuer Software, Änderungen der Besitzer einer Datei oder Verwaltung von anderen Benutzerkonten.

Das Root-Konto ist nicht für die alltägliche Verwendung des Systems gedacht wie zum Beispiel das Surfen im Web, Texte schreiben usw., sondern rein für die

Verwaltungsaufgaben. Denn ein einfacher Fehler kann dann zu Problemen mit dem gesamten Linux System führen, da der Root-User ja alle Berechtigungen hat.

Wenn man für spezielle Aufgaben Root-Rechte benötigt dann kann man auch den „**sudo**“ Befehl anwenden. Dies stellt dann sicher das nur dieser eine Befehl mit Root-Rechten ausgeführt wird. Für die Einrichtung und Konfiguration des Server empfiehlt es sich auch direkt zum Root-User zu Switchen mit „**su**“

#### **Hinweis:**

In einigen Linux Distributionen wie zum Beispiel bei Ubuntu wird dem Root-User **kein Passwort** zugewiesen. Dadurch kann sich niemand unter dem Namen „**root**“ anmelden.

### 7.4.2 Neuen Benutzer anlegen mit „adduser“

Benutzer anlegen kann man entweder über die grafische Oberfläche, welche bei jeder Distribution anders aussehen kann oder direkt über das Terminal, welches ich Ihnen hier zeigen werden, da es Großteils Linux Distribution unabhängig ist.

Einen neuen Benutzer können wir anlegen mit „**adduser**“ und dem „**sudo**“ Befehl:

**# sudo adduser neuerbenutzername**

Im Anschluss wird man neben dem Passwort einiges gefragt was man nicht unbedingt angeben muss, wie zum Beispiel der Name, Telefonnummer usw.

#### **Hinweis:**

In manchen Linux Distributionen kann es vorkommen das „**adduser**“ auch „**useradd**“ benannt wird. In der Regel wird man jedoch „**adduser**“ vorfinden.

### 7.4.3 Neuen Benutzer löschen mit „deluser“

Wenn wir Benutzer anlegen müssen wir auch wissen wie wir Benutzer wieder löschen können. Hierzu gibt es den Befehl „**deluser**“ welchen wir wieder mit Root-Rechten also „**sudo**“ ausführen können:

**# sudo deluser neuerbenutzername**

#### **Hinweis:**

Um das komplette Homeverzeichnis sowie alle zugehörigen Mails des Benutzers zu löschen, benutzt man die Option „**-remove-home**“, dies kann man auch nachträglich händisch löschen.

#### **Hinweis 2:**

Wie auch bei „**adduser**“, gibt es auch bei „**deluser**“ einige Linux-Distributionen welche den Befehl umgedreht haben und stattdessen „**userdel**“ bevorzugen. In der Regel wird man jedoch „**deluser**“ benutzen,

### 7.4.4 Gruppen anlegen mit „addgroup“

Mit dem Befehl „**addgroup**“ kann man in Linux neue Gruppen anlegen. Wir müssen nur den Namen der Gruppe noch definieren. Alternativ können wir auch noch zusätzlich die „**GID**“,

eine sogenannten „**Group ID**“ mit angeben. Falls wir keine gewünschte GID angeben wird die nächste freie verfügbare GID verwendet.

Hier ein Beispiel einer neuen Gruppe namens „freunde“:

```
# sudo addgroup freunde
```

Und nun das Beispiel mit einer GID in dem Falle nutze ich die GID 2000:

```
# sudo addgroup --gid 2000 freunde
```

Benutzer einer Gruppe hinzufügen kann man wieder mit dem Befehl „**adduser**“, hierzu fügen wir den Benutzer „**rene**“ in die Gruppe „**freunde**“ hinzu:

```
# sudo adduser rene freunde
```

#### 7.4.5 Gruppen löschen mit „delgroup“

Genauso wie wir Gruppen anlegen, können wir auch Gruppen wieder löschen. Hierzu gibt es den Befehl „delgroup“.

**Anmerkung:** Sie können eine primäre Gruppe eines vorhandenen Benutzers nicht löschen. Hierzu müssen Sie zu erst den Benutzer löschen oder seine primäre Gruppe verändern.

Ein Beispiel wie wir unsere erstellte Gruppe „**freunde**“ löschen:

```
# sudo delgroup freunde
```

#### Hinweis:

Wir können auch die Gruppen direkt bearbeiten indem wir die Datei „**/etc/group**“ bearbeiten. Dies kann ich jedoch nicht empfehlen, außer Sie wissen genau was Sie tun. Denn nur „**delgroup**“ überprüft vorher ob ein Benutzer noch in dieser als primäre Gruppe eingetragen ist.

## 7.5 Verwaltung des Betriebssystems

### 7.5.1 Cronjob - Wiederkehrende Aufgaben

Der Cron bzw. Cron-Daemon ist ein Dienst, der automatisch Skripte und Programme zu vorgegeben Zeiten starten kann. Mittels der „crontab“ Tabelle können wir diese Zeiten definieren. Es gibt hierfür eine Datei „**/etc/crontab**“, welche jedoch nur mit Root-Rechten bzw. Sudo bearbeitet werden kann. Zusätzlich zur Systemweiten Crontab können Benutzer auch eine eigene „crontab“ erstellen, diese befindet sich dann im Verzeichnis „**/var/spool/cron/crontabs/**“.

Um einen neuen cronjob über das Terminal einzurichten benötigen wir das Programm „crontab“ mit der Option -e zum editieren.

```
# crontab -e
```

Aus historischen Gründen wird ein crontab mit einem editor wie „nano“ bearbeitet.

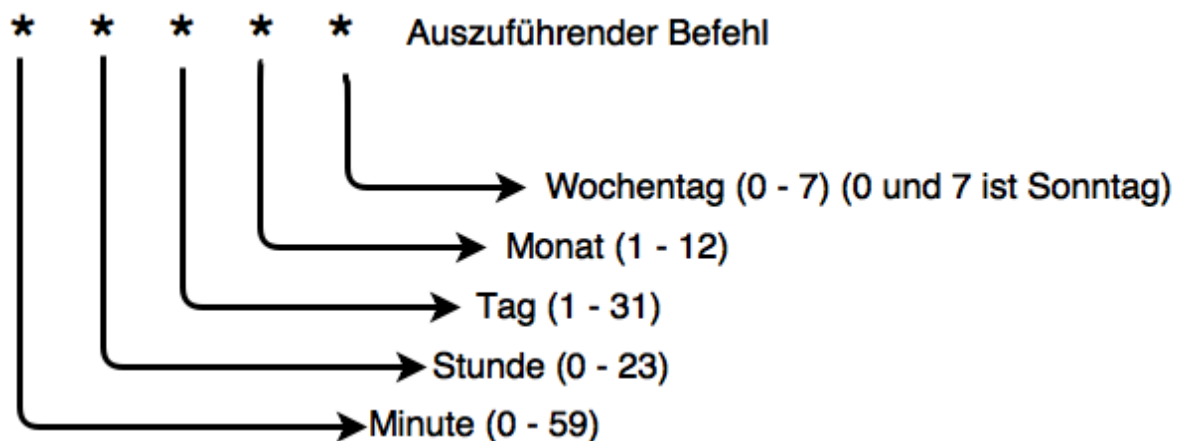
Als Cron Beispiel überprüfen wir ob ein Server oder Internetseite erreichbar ist und im Falle eines Fehlers wir eine E-Mail bekommen. Dazu „pingen“ wir alle 5 Minuten eine Webseite.

Der Cronjob Befehl hat dann wie folgt auszusehen:

```
*/5 * * * * ping -c 1 www.thinksecret.at > /dev/null 2>&1
```

Darüber hinaus definieren wir noch die E-Mail adresse mit MAILTO=""  
MAILTO="ihre@emailadresse"

Die Zeiten für den Cronjob werden durch die Sterne definiert, dabei beginnt der erste Stern bei Minute und der letzte beim Wochentag.



Angaben von Zeiträumen und Schrittweiten innerhalb sind erlaubt. Setzt man als Beispiel Stunde „5-8“, so wird der Befehl in der Stunde 5,6,7 und 8 ausgeführt. Alternativ könnte man auch „5,6,7,8“ schreiben. Wenn man hingegen „\*/5“ wie bei der Minute schreibt, so wird der Befehl alle 5 Minuten durchgeführt, da alle anderen Werte als \* deklariert wird.

### 7.5.2 Instanzen und Prozesse in Linux

Eine Instanz eines laufenden Programmes wird als Prozess bezeichnet. Jedes mal, wenn wir ein Terminal öffnen oder irgendein anderes wird dafür ein Prozess geschaffen. Jeder Prozess hat eine eigene Prozess-ID (**PID**) und ist einem bestimmten Eigentümer zugewiesen.

Jedes moderne Betriebssystem ist ein Multitasking-Betriebssystem, damit ist gemeint, dass mehrere Programme gleichzeitig ausgeführt und verarbeitet werden können. Unter Windows kann man diese Prozesse als „**Tasks**“ und sind im Taskmanager sichtbar. Alle Prozesse miteinander teilen sich die gesamten Hardware Ressourcen wie CPU und Speicher.

### 7.5.3 Laufende Prozesse anzeigen mit „ps“

Eine Übersicht über alle Prozesse können wir uns mit dem Befehl „**ps**“ anzeigen lassen. Wenn Sie „**ps**“ im Terminal eingeben werden Ihnen nur Prozesse angezeigt welche von der aktuellen Session aus gestartet wurden. Um Informationen über alle Prozesse auf dem System zu bekommen führen sie die Option „**-a**“ für „**all**“ aus:

```
# ps -a
```

Die am häufigsten verwendeten Optionen mit ps sind „a“, „u“, „x“. Diese Optionen zeigen alle laufenden Prozesse für alle Benutzer, sowie nützliche Informationen wie **PID**, Benutzernamen, CPU-Last, Startzeit des Prozesses, Befehl welchen den Prozess gestartet hat:

# ps -aux

#### 7.5.4 Laufende Prozesse mit „top“

Der Befehl „top“ wird genau wie „ps“ verwendet. Doch „top“ hat noch eine sehr nützliche Eigenschaft: Die Anzeige wird grafisch aufbereitet und alle paar Sekunden aktualisiert.

# top

In „top“ können sie auch noch weitere Einstellungen vornehmen und die Sortierungen ändern. Meistens wird dies genutzt um Prozesse zu erkennen welche eine hohe CPU, RAM Auslastung haben.

Wenn Sie sich in „top“ befinden können sie mit „M“ die Sortierung nach der Speichernutzung anzeigen lassen. Für die Sortierung der CPU-Auslastung drücken sie einfach „P“. Sie können auch nach allen anderen Feldern sortieren lassen, hierfür springen Sie einfach innerhalb der Spalten mit „<“ nach links, und mit „>“ nach rechts hin und her. Um in die Hilfe zu kommen drücken Sie einfach „h“.

#### 7.5.5 Laufenden Prozesse beenden mit „kill“

Mit dem „kill“ Befehl können wir Prozesse beenden. Wenn ein Programm nicht mehr reagiert oder hohe CPU-Last verursachen ist es wahrscheinlich das sich dieser Prozess aufgehängt hat und sich selbst nicht mehr beenden kann. Wir können mit dem „kill“ befehl Signale an den Prozess schicken und ihn nicht nur dazu zwingen zu beenden, sondern auch die Möglichkeit sich neu zu starten und zum Beispiel Konfigurationsdateien neu einzulesen. Die Signale werden von verschiedenen Zahlen dargestellt. Wenn kein Signal angegeben wird, dann wird der Standardwert 15 übergeben, was bedeutet das der Prozess beendet werden soll, jedoch nur wenn alle Aufgaben erledigt worden sind. Bei hängenden Prozessen nützt dies nichts da die Aufgaben wie beschrieben „hängen“.

Signalliste für den Kill Befehl:

Signal	Nummer	Bedeutung
<b>SIGHUP</b>	1	Blockierung des Kontrollterminals
<b>SIGKILL</b>	9	Sofortiges Beenden
<b>SIGTERM</b>	15	Anfrage zum Beenden
<b>SIGSTOP</b>	17,19,23	Pausiere Prozess
<b>SIGCONT</b>	18,19,25	Fortfahren Prozess

Um nun ein sofortiges beenden einzuleiten nutzen wir meistens die Signalnummer „9“, hierzu müssen wir zuerst die **PID** (Prozess-ID) ermitteln welchen wir beenden wollen.

# ps -aux

Als Ausgabe erhalten wir eine lange Liste mit den zugehörigen PID, ein Beispiel:

# www-data **30876** 0.0 0.4 384448 22244 ? S 15:50 0:01 /usr/sbin/apache2

Nun wollen wir den Apach2 Prozess beenden und nutzen hierfür die PID:  
**# kill -9 30876**

Wenn Sie jetzt nochmals ein „**ps -aux**“ durchführen werden Sie sehen das der Prozess beendet wurde.

Apache ist unser Webserver, der im Normalfall viele Prozesse auf einmal startet. Damit wir nicht jede PID einzeln beenden müssen gibt es die Methode von „**killall**“. Sie ist ähnlich aufzuführen wie kill, nur das wir jetzt den Namen des Prozesses angeben.  
**# killall -9 apache2**

Nun wurden alle apache2 Prozesse beendet und wir können dies wieder überprüfen mit „ps -aux“. Hin und wieder kann es vorkommen das Prozesse noch weiterlaufen, in diesem fälle einfach mehrmals den „killall“ Befehl ausführen.

#### 7.5.6 Offizielle Software Installieren mittels Paketmanager

Ein Paketmanager ist eine Sammlung von Software, welche den Prozess der Installation, Aktualisierung, Konfiguration und Deinstallation von Software auf einem System automatisieren. Vergleichbar mit einem Appstore auf dem Handy oder unter Windows 10 gibt es dies unter Linux schon seit über 15 Jahren.

Mit einem einzigen Befehl führen Sie also den gewünschten Download und Installation aus.

Verschiedene Linux Distributionen nutzen verschiedene Paketmanager. In der Regel werden Sie auf den „**RPM**“ (**Red Hat Derivate**) **Paketmanager** und dem „**DEB**“ (**Debian Derivate**) **Paketmanager** stoßen.

#### 7.5.7 Red Hat Derivat Software Installieren, Deinstallieren mit RPM

Einige RPM-basierende Distributionen sind Red Hat, Fedora, CentOS, SUSE Enterprise, OpenSuse und Mandriva. RPM wird verwendet um Software in diesen Red Hat Derivaten zu Installieren, Überprüfen, Aktualisieren und Deinstallieren. Durch die Verwendung von „**rpm**“ mit den zugehörigen Optionen, kann man alle Aktionen durchführen die man möchte.

Auf der Webseite von [www.rpmfind.net](http://www.rpmfind.net) finden Sie eine Übersicht über alle Verfügbaren Pakete und können diese dort direkt Downloaden.

Die Wichtigsten Optionen des RPM Paketmanagers sind:

Option	Beschreibung
-i	.rpm Paket installieren
-U	.rpm Paket aktualisiere (Update)
-q	RPM Datenbank nach installierten Paketen abfragen
-e	.rpm Paket entfernen
-v	Detaillierte Informationen anzeigen

Die meisten RPM Dateien beinhalten die Versionsnummer und Buildnummer im Dateinamen.

Beispiel zur Installation eines RPM Paketes von Apache 2 (Version 2.4.16 Build 9.1):

```
# sudo rpm -iv apache2-2.4.16-9.1.x86_64.rpm
```

Deinstallation von Apache2 können wir mit „-e“ vollziehen:

```
# sudo rpm -e apache2-2.4.16-9.1.x86_64.rpm
```

#### 7.5.8 Debian Derivat Software mit APT

In der Regel verwendet man bei Ubuntu und anderen Debian Derivaten den sogenannten APT (Advanced Packaging Tool), welche den ganzen Installationsprozess sehr vereinfacht. Ursprünglich wurden APT entwickelt als ein Frontend für den „dpkg“ (Debian Paketmanager). APT besteht aus riesigen Programmdatenbanken womit Programme direkt online abgerufen und einfach Heruntergeladen, Installiert und Vorkonfiguriert werden.

Die häufigsten verwendeten Programme sind:

„apt-cache“, welche Informationen aus der Datenbank ausliest, welche Programme es gibt.  
„apt-get“, wird für Installation, Aktualisierung und Deinstallation von Programmen genutzt.

Im jetzigen Beispiel installieren wir Apache 2 (Webserver) auf unser Linux System.

#### 7.5.9 Pakete suchen mit „apt-cache“

Mit dem Befehl apt-cache können wir nach Paketen suchen und uns weitere Informationen und Statistiken anzeigen lassen. Informationen werden vom Befehl „**apt-get update**“ auf den Online quellen gesammelt welche in der „**sources.list**“ angeführt ist.

Wir durchsuchen die Datenbank nach Apache 2 Paketen:

```
# sudo apt-cache pkgnames apache2
```

oder

```
# sudo apt-cache search apache2
```

Die Ausgabe ist „**Paketname** - Beschreibung:

**apache2** - Apache HTTP Server metapackage

Wollen wir nur weitere Informationen über das Paket „apache2“ haben nutzen wir showpkg

```
# sudo apt-cache showpkg apache2
```

Abhängigkeiten von Paketen anzeigen, denn oft nutzen Pakete andere Pakete welche voneinander abhängig sind:

```
# sudo apt-cache depends apache2
```

Statistiken über die Pakete anzeigen:

```
# sudo apt-cache stats
```

### 7.5.10 Pakete installieren mit „apt-get“

„apt-get“ ist das Programm für die Installation und Verwaltung von Debian Paketen. Das Programm greift auf die Online ressourcen zu welche in der Datei **„/etc/apt/sources.list“** aufgeführt sind.

Um die Liste der Pakete zu aktualisieren, müssen wir die **„sources.list“** neu einlesen:

```
# sudo apt-get update
```

Um alle Pakete auf die aktuellste Version zu aktualisieren können wir die Option „-u“ verwenden mit „upgrade“:

```
# sudo apt-get -u upgrade
```

**Hinweis:** Es ist immer wichtig **„apt-get update“** vor **„apt-get upgrade“** zu benutzen.

**„dist-upgrade“** bringt die installierten Pakete auf den neuesten Stand. Hierbei werden im Gegensatz zu upgrade neue Pakete installiert und durch neue Abhängigkeiten unnötig gewordene Pakete ersetzt, auch wenn dies alte Abhängigkeiten beeinflusst. **„dist-upgrade“** benötigt keine Angabe von Paketnamen

#### Einzelnes Paket installieren

Das „apache2“ Paket wollen wir nun installieren:

```
# sudo apt-get install apache2
```

#### Paket-Cache leeren

Mit **„apt-get clean“** können wir das lokale Verzeichnis, in dem sich die Pakete welche wir Installiert haben, löschen.

**Hinweis:** „apt-get“ Aktionen wie die Installation und Deinstallieren von Paketen werden in einer Log Datei **„/var/log/dpkg.log“** protokolliert.

APT bietet noch viel mehr Funktionen, die wichtigsten haben wir hier durchgemacht. Weitere Informationen zu APT findet man in den einzelnen **„man“** Pages.

## 8. Zusammenfassung

### 8.1.1 Gratulation zum Linux für Anfänger Kurs

Wir sind nun am Ende diese E-Books angelangt. Hoffentlich haben Sie viel dabei gelernt und Ihr Wissen über Linux vorangebracht. Wenn Ihnen diese E-Book gefallen hat würde ich mich über ein Feedback freuen oder eine Teilnahme an meinen Videokurs, um die Tutorials zu unterstützen.

## 8.2 Die wichtigsten Webseiten zum Thema

### 8.2.1 Linux Distributionen und Tools

Debian: <http://www.debian.org>

Ubuntu: <http://www.ubuntu.com>

CentOS: <http://www.centos.org>

openSUSE: <http://www.opensuse.org>

Fedora: <http://www.fedoraproject.org>

Virtual Box: <http://virtualbox.org>

Putty (freier SSH Client für Windows): <http://www.putty.org>

### 8.2.2 Die Besten und günstigsten Root und Virtuellen Server

Virtuelle Server ab 5 € / Monat:

DigitalOcean: <http://thinksecret.at/DigitalOcean> (inkl. 10 \$ Startguthaben)

Root-Server für Private und Test Umgebungen ab 6 € / Monat:

1. Kimsufi (Frankreich): <http://www.kimsufi.com>
2. SoYouStart (Frankreich): <http://www.soyoustart.com>
3. Hetzner (Deutschland): <http://www.hetzner.de>

Root-Server für klein und mittlere Unternehmen ab 24 € / Monat:

1. Hetzner (Deutschland): <http://www.hetzner.de>
2. OVH (Frankreich): <http://www.ovh.com>
3. Server4You: <http://www.server4you.de>

Root-Server für große Unternehmen ab 84 € / Monat

1. OVH (Frankreich): <http://www.ovh.com>

### 8.2.3 Domain Registrierung und Webhosting

1. Easyname (Österreich): <http://www.easyname.eu>
2. Domaindiscount24 (Deutschland): <http://www.domaindiscount24.com>

### 8.2.4 Wordpress

Wordpress: <http://www.wordpress.org>

Wordpress Themes: <https://de.wordpress.org/themes>

Wordpress Plug-Ins: <https://de.wordpress.org/plugins/>

### 8.2.5 Videokurse

Tutorials zu allem Möglichen gibt es auch bei Youtube und Tutkit

<https://www.youtube.com/channel/UCW4eOhtyq3kl36f9-KIdpEA>

<https://www.tutkit.com/trainer/rene-fuerst--81>

### 8.2.6 Autorenwebseite

Mein persönlichen Blog und weitere Informationen zu mir und meinen Kursen finden Sie unter: <http://www.thinksecret.at>