

Kursübersicht

June 26, 2019

1 Einführung in Linux

Lange Zeit musste ein Nutzer, der mit Linux arbeiten wollte, ein Linux-Betriebssystem parallel zu Windows installieren und beim Hochfahren des Computers auswählen welches Betriebssystem, Windows oder Linux, er nun gerade benutzen möchte.

Durch die Software [VirtualBox](#) kann man heute einfach eine (oder auch mehrere) Linux-Betriebssysteme als untergeordnete Programme in Windows installieren.

1.1 Installation

- Laden Sie sich zunächst die ISO-Datei der Linux-Distribution Ihres Vertrauens von der jeweiligen Website herunter. (Hinweis: Laden Sie sich immer möglichst die Desktop-Version, nicht die Server-Version herunter.)
 1. [Ubuntu LTS](#)
 2. [Linux Mint Cinnamon](#)
 3. [Zorin OS](#)
 4. [Manjaro](#)
- Installieren Sie VirtualBox.
- Öffnen Sie VirtualBox und klicken auf "Neu".
- Wählen Sie nun einen Anzeigenamen für Ihr Linux (z.B. "Ubuntu"). Die anderen Einstellungen versucht VirtualBox möglichst automatisch zu konfigurieren.
- Als nächstes können Sie die Grösse des Arbeitsspeichers für Linux bestimmen. Lassen Sie es fürs erste bei den Standardeinstellungen. Sie können dies jederzeit später ändern.
- Nun können Sie die Festplattengrösse für Ihr Linux einstellen. Vergessen Sie nicht, dass dies der Platz auf Ihrer tatsächlichen Festplatte ist! Wählen Sie "Festplatte erzeugen".
- Bei den folgenden Schritten können Sie einfach die Standardeinstellungen wählen. Die Festplattengrösse für Ihr Linux ist standardmässig auf 10 GB eingestellt. Das ist gut zum Testen, aber fürs Arbeiten zu wenig! Spätere Änderungen sind nur sehr schwer möglich. Sie sollten deshalb hier eine ausreichende Grösse wählen, die nicht gleichzeitig den kompletten freien Speicher des Computer auffrisst.

- Nun können Sie auf “Erzeugen” klicken und kommen zurück ins Startmenü. Dort wurde Ihre Linux-Distribution in die Auswahlliste eingefügt. Mit einem Doppelklick darauf können Sie Linux nun starten.
- Da noch kein Linux auf der Partition installiert ist, wird beim ersten Start ein Fenster geöffnet, das Sie bittet, dass Sie einen Datenträger für die Installation auswählen. Hier wählen Sie nun die heruntergeladene ISO-Datei Ihrer Linux-Distribution aus.
- Anschließend sollte entweder die Installation starten oder Linux direkt auf die Desktop-Oberfläche booten (abhängig von der Distribution).

1.2 Unterschiede von Linux und Windows

Eines vorweg: Keines der beiden Betriebssysteme ist dem anderen überlegen. Beide haben Ihre Vor- und Nachteile. Für den normalen Nutzer ergibt sich, abgesehen von der Optik, wenige Unterschiede. Beide Betriebssysteme haben ein Office-Programm (MS-Office vs. LibreOffice), Musikplayer, Browser etc..

Der Hauptunterschied findet sich in der Nutzbarkeit der Terminals (auch “Konsole” genannt) - also das berühmte schwarze Fenster mit dem grünen, blinkenden Cursor. Das Terminal erlaubt viele Dinge (auch alltägliches) schnell zu erledigen. Ein Beispiel: Die Installation von GIMP (die kostenlose Alternative zu Photoshop).

Unter Windows müssen Sie auf die GIMP-Homepage, laden die exe-Datei herunter und installieren das Programm mit vielen Klicks durch diverse Menüs. Und dann kann es passieren, dass die GIMP-Homepage zuvor gehackt wurde und Sie sich gerade einen Virus heruntergeladen haben ohne es zu merken.

Unter Linux öffnen Sie das Terminal, geben dort ein `apt install gimp` und die Installation läuft. Und Viren sind hoch unwahrscheinlich, da die Software nur von vertrauenswürdigen Servern heruntergeladen wird.

Da sich der Arbeitsalltag durch das Terminal stark vereinfacht lässt, wird der Hauptfokus des Kurses eben auf dem Terminal liegen - weil viele Dinge doch mittlerweile gleich sind zwischen Linux und Windows.

1.3 Grundlagen des Linux-Terminals

Sobald Sie das Terminal von Linux öffnen, steht Ihnen das komplette Betriebssystem offen. Sie können nun wirklich alles machen. Windows erlaubt Ihnen diese Freiheit nicht. Aber bevor man ALLES machen kann, muss man erstmal wissen wie man ETWAS macht.

Fangen wir also ganz simpel an. Lassen wir uns den Inhalt des Ordners anzeigen, in dem wir uns gerade befinden.

[2]: `ls`

`Kursübersicht.ipynb`

`ls` steht einfach für ‘list’ - einfach weil Linux-Nutzer faul sind und nicht jedesmal ‘list’ eintippen wollen. Der Befehl gibt Ihnen den aktuellen Inhalt (Dateien und Ordner) des Ordners, in dem Sie sich befinden, aus (beim Start der Konsole befinden Sie sich immer in Ihrem Benutzerverzeichnis). Falls Ihnen die Informationen durch `ls` nicht reichen, können Sie noch Parameter mitgeben, welche den Befehl nach Ihrem Wünschen anpassen.

```
[3]: ls -l
```

```
total 16
-rw-r--r-- 1 apachzelt  staff  5942 26 Jun 09:57 Kursübersicht.ipynb
```

Parameter werden meist mit einem oder zwei Bindestrichen an den Befehl angehängt. Hier hat das `-l` dafür gesorgt, dass wir weitere Details für jedes Element im aktuellen Ordner bekommen. Das sieht nun etwas kryptisch aus: `* -rw-r--r--` = Berechtigungen (Read / Write / eXecute) für den Dateibesitzer, -gruppe und Andere `* apachzelt staff` = Der vordere Name ist der Besitzer, der hintere der einer Gruppe von Nutzern `* 5942` = Dateigröße in Bytes `* 26 Jun 09:57` = Änderungsdatum `* Kursübersicht.ipynb` = Datei-/Ordnername

So gut wie jeder Befehl im Terminal akzeptiert Parameter. Um eine Übersicht der Parameter inkl. Erklärung zu bekommen, kann man einfach den Handbucheintrag des Befehls nachlesen.

```
[ ]: man ls
```

An man können Sie jeden Befehl anhängen und erhalten so einen Einblick in die jeweilige Funktion.

Nun wollen wir aber nicht ewig im Benutzerverzeichnis verweilen, sondern wollen auf den Desktop. Unter Ubuntu ist dies der "Schreibtisch". Und er ist auch vom Terminal erreichbar.

```
[ ]: cd Schreibtisch
```

`cd` bedeutet "change directory" (auch hier wieder: Faulheit!). Mit `cd` wechseln Sie von einem in den nächsten Ordner. Wenn Sie wieder einen Schritt zurück wollen, tippen Sie einfach `cd ..`. Das `..` bedeutet im Terminal "übergeordneter Ordner".

Nun können Sie mit `ls` wieder alle Dateien auf Ihrem Desktop sehen. Für die weitere Arbeit brauchen wir aber einen Ordner auf dem Desktop. Diesen Erstellen wir mit dem Befehl `mkdir` ("make directory").

```
[17]: mkdir Workshop
```

```
[18]: ls
```

Workshop

Im gezeigten Beispiel liegt nur unser gerade erstellter Ordner auf dem Desktop.

Packen wir nun einige Beispieldateien in den Ordner "Workshop" (nicht gezeigt) und wechseln in den Ordner.

```
[20]: cd Workshop
ls
```

Bilder PDF

Scripts

Sie sehen, dass im Ordner "Workshop" nun 3 Ordner liegen. Wir können uns mit `ls` auch den Inhalt von Unterordnern anzeigen lassen.

```
[21]: ls PDF/
```

[Biologie](#)
[Geisteswissenschaften](#)
[Gebäude](#) [Personen](#)

Im Ordner "PDF" liegen also 4 Ordner. Was wohl im Ordner "Biologie" drin ist?

```
[22]: ls PDF/Biologie/
```

```
Anthropologie.pdf  Pflanzenfresser.pdf
```

Im Ordner "Biologie" liegen 2 pdf-Dateien. Öffnen wir nun beispielhaft "Anthropologie.pdf".

```
[ ]: less PDF/Biologie/Anthropologie.pdf
```

Wie Sie sehen, können Sie die Datei "Anthropologie.pdf" öffnen ohne in den Ordner, in dem die Datei liegt, zu wechseln. Der Befehl `less` ist ein einfaches Programm, das Ihnen nur erlaubt den Inhalt von Dateien anzusehen aber nicht zu bearbeiten. Das funktioniert natürlich am Besten mit Textdateien. Andere Dateien werden auch angezeigt - der angezeigte Inhalt muss aber nicht unbedingt Sinn machen.

Sie verlassen `less` übrigens durch Betätigen der Taste `q` ("quit").

Nehmen wir nun an, dass wir in den PDFs ein Wort suchen, z.B. "Arbeit". Dazu können wir den Befehl `grep` nutzen.

```
[9]: grep -ir arbeit PDF/
```

```
Binary file PDF//Geisteswissenschaften/Ökotropologie.pdf matches  
Binary file PDF//Geisteswissenschaften/Sozialmedizin.pdf matches
```

`grep` durchsucht den INHALT von Dateien nach dem gegebenen Suchbegriff. Dazu haben wir 2 Parameter mitgegeben: `-i` und `-r` (der Einfachheit halber kann man mehrere Parameter einfach hintereinander schreiben: `-ir`). `i` steht für "case-insensitive", ignoriert also Groß- und Kleinschreibung. `r` heißt "rekursiv" und bedeutet, dass nicht nur der gegebene Ordner durchsucht wird, sondern auch alle Unterordner.

Und dann gibt uns der Befehl aus, dass 2 PDFs den gesuchten Begriff enthalten.

1.4 Automatisierung unter Linux

Um sich den Alltag mit Linux einfacher zu machen, können Sie Befehle oder selbstgeschriebene Skripte in regelmäßigen Abständen automatisch laufen lassen (sogenannte "cron jobs").

1.4.1 Nach Suchbegriff in vielen PDFs suchen

Aber auch einmalige Arbeiten können mit dem Terminal beschleunigt werden. Stellen Sie sich vor, Sie wollen alle PDFs in einem Ordner nach einem Stichwort durchsuchen und alle Dateien mit einem Treffer in einen anderen Ordner sammeln und in Textdateien konvertieren, damit Sie diese in ein anderes Programm einbinden können, das keine PDFs liest.

Das ist selbst mit der Konsole einiges zu schreiben. Und falls Sie sich dann später entscheiden, dass Sie doch nach einem anderen Stichwort suchen wollen, müssen Sie den ganzen Prozess wiederholen. Da kommen Skripte ins Spiel. In einem Skript sind einfach mehrere Terminal-Befehle untereinander geschrieben. Und diese werden dann nacheinander ausgeführt.

Beispielskripte finden Sie im Ordner "Skripte". Für das gegebene Beispiel hilft uns das Skript "extractTextFromPDF.sh". Falls Sie das Skript genauer betrachten wollen, können Sie es mit `less Scripts/extractTextFromPDF.pdf` öffnen.

Zum Ausführen geben Sie dem Skript einfach einen Suchbegriff:

```
[13]: Scripts/extractTextFromPDF.sh arbeit
```

Das Skript macht nur seine Arbeit und schreibt Ihnen nichts aufs Terminal. Dennoch können Sie nun sehen, dass es in Ihrem aktuellen Ordner einen neuen Ordner "script-output" geben sollte, in dem sich 2 Textdateien befinden, die wir zuvor schon mit dem Suchbegriff gefunden haben - nur als txt-Dateien.

Sie könnten dem Skript auch noch den Ordner, in dem es suchen soll, und den Ordner, in den es schreiben soll, mitgeben.

```
[15]: Scripts/extractTextFromPDF.sh arbeit PDF/Geisteswissenschaften Ausgabe
```

Das Skript erwartet diese Angaben EXAKT in dieser Reihenfolge - Suchbegriff, Eingangsordner, Ausgangsordner. Sie können den Ausgangsordner weglassen, aber nicht den Eingangsordner und den Ausgangsordner angeben. `Scripts/extractTextFromPDF.sh arbeit Ausgabe` würde also nicht zu dem gewünschten Ergebnis führen. Vielmehr würde das Skript den Ordner "Ausgabe" nach dem Suchbegriff durchsuchen.

1.4.2 Vollautomatische Konvertierung von Dateiformaten

Manchmal wollen Sie aber nicht jedes Mal das entsprechende Skript im Terminal ausführen. Sie wollen, dass eine Aufgabe wirklich automatisch erledigt wird.

Dazu installieren wir zunächst zwei zusätzliche Programme.

```
[ ]: sudo apt install inotify-tools pandoc
```

Der Befehl `apt install` installiert die beiden Programme `inotify-tools` und `pandoc`. Installieren können Sie unter Linux aber nur als Administrator (im Linux-Jargon auch "root" genannt). Um mit Administrator-Rechten einen Befehl auszuführen, beginnen Sie einen Befehl mit `sudo` ("super-user do").

Das Programm `inotify-tools` überwacht gegebene Ordner auf Dateien oder Ordner, die hineingelegt werden oder sich verändern (also bearbeitet werden).

Legen wir also zunächst auf dem Desktop zwei neue Ordner an - "Eingang" und "Ausgang" (achten Sie auf die GroSS- und Kleinschreibung).

```
[19]: cd ~/Schreibtisch/  
mkdir Eingang Ausgang
```

Für die folgende Aufgabe nehmen wir an, dass wir eine Zeitschrift herausgeben und jeden Tag zahlreiche Artikel bei uns eingereicht werden. Leider nutzen viele Wissenschaftler immer noch Word, um Artikel zu schreiben. Wir benötigen unsere Artikel aber in anderen Formaten: LateX, PDF und HTML. Daher wollen wir einen automatischen Workflow etablieren, der uns Word-Dateien automatisch in die genannten Formate konvertiert.

Dabei hilft uns das Skript `convertWordFiles.sh`. Einmal gestartet, überwacht es den Ordner "Eingang". Sobald dort eine Word-Datei hineingelegt wird, startet es automatisch die Konvertierung in die genannten Formate und legt sie im Ordner "Ausgang" ab.

```
[ ]: Scripts/convertWordFiles.sh
```

1.4.3 Automatische Bearbeitung von Bildern

Manchmal müssen Bilder nach einem bestimmten Muster bearbeitet werden. Und sobald die Zahl der Bilder in die Hunderte geht, kann es praktisch sein ein Skript zu haben, dass dies erledigt.

Im Ordner "Bilder" finden Sie Unterordner von zwei Jahrgängen einer Zeitschrift. Darin befinden sich Scans von Buchseiten - pro Bild eine Seite. Jede Seite hat aber einen schwarzen Rand. Dieser Rand ist aber nicht gewünscht; er muss entfernt werden.

Linux ist in der Lage Bilder im Terminal zu bearbeiten. Es nutzt dazu das Programm `imagemagick`.

Auch hier liegt ein passendes Skript `cutImages.sh` im Ordner "Skripte". Es startet in einem Ordner und sucht rekursiv (also mit allen Unterordnern) nach Bilddateien, die es dann nach einem bestimmten Muster beschneidet und als `jpg`-Datei im Ordner darüber speichert.

```
[ ]: cd ~/Schreibtisch/Workshop/Bilder  
    ../Scripts/cutImages.sh
```

```
[ ]:
```