

Linux Kurs - Überblick

Franz Schäfer

2019-01-11

1 Was ist “Linux”? Was ist “GNU”? Was ist “Freie Software”? (10 Min)

1.1 Was ist Linux?

Linux ist ein freies **Betriebssystem**. Ein Betriebssystem kümmert sich um den Zugriff auf die Hardware. Es verwaltet den Speicher, es “weiß” wie man Daten auf eine Festplatte schreibt, wie die Netzwerkkarte angesprochen werden will, etc, etc.

Zum Betriebssystem gehört der so genannten “Kernel”, der die oben erwähnten Aufgaben wahrnimmt und zusätzlich auch noch mehr oder weniger viele System und Anwenderprogramme die das ganze drum herum regeln. Z.b. Wie das verarbeiten eingegebener Befehle.

Das Betriebssystem bietet also Schnittstellen für andere Programme die dort laufen können (eine so genannte API) und eine Schnittstelle für die BenutzerInnen (z.b. Befehlseingabe und optional eine graphische Oberfläche).

Die Schnittstellen von Linux entsprechen denen von Unix. Vom GNU Projekt wurden viele Freie Unix Programme geschaffen auf denen die dann Linux aufbauen konnte. GNU ist eine rekursive Abkürzung für “GNU is Not Unix”.

1.2 Was ist “Freie Software”? Was ist “Open Source”?

“Freie Software” und “Open Source” werden oft synonym benutzt. Richard Stallman, Vordenker der Bewegung bevorzugt den Begriff “Freie Software” um auf den politischen Aspekt hinzuweisen. “Open Source” wird i.a. von der libertären Seite benutzt: Die den Vorteil von Software, die mit Source Code, den man/frau auch selbst Verändern kann, zwar schätzen aber dies nicht als politisches Statement sehen wollen.

Als politisch aktive Menschen sollten wir daher eher den Begriff “Freie Software” verwenden, auch wenn der Begriff “Open Source” weiter verbreitet ist. Als Kompromiss bietet sich FLOSS an: “Free Libre Open Source Software”.

Die Frage ist oft auch eine praktische: Wann ist software “frei”?

Z.b: Manche Firmen geben ihre Software mit Source Code weiter, erlauben aber dessen Verwendung nur sehr eingeschränkt.

Eine praktische Definition zur Unterscheidung der sehr vielen Software Lizenzen wurde mit den “Debian Free Software Guidelines (DFSG)” geschaffen.

Für Richard M. Stallman (RMS) ist das “frei” wie in “Freiheit” und nicht wie in “Freibier” gemeint. Es geht um unsere Rechte.

Eine der wichtigsten Software Lizenzen ist die von Stallman geschaffene GPL.

2 Was sind die Vorteile von Linux und von “Freier Software”? (10 Min)

relativ zukunftssicher solange sich Menschen finden die ein wenig Ahnung von Software haben und die das Programmpaket nützen wollen, wird es das Paket weiter geben. Es kann nicht so einfach “vom Markt verschwinden”.

erlerntes know-how bleibt weiter nützlich Wer ein Unix von vor 40 Jahren genutzt hat wird sich auch auf einem modernen Linux gut zurecht finden. Selbiges gilt auch für die meisten Programme. z.B.: \LaTeX

alternative zum Kapitalismus für alle die auf ein relativ gut funktionierendes Beispiel von freier Kooperation verweisen wollen, ist Freie Software ein sehr gutes Beispiel.

Die UserInnen werden nicht eingesperrt Kommerzielle Programme zielen oft darauf ab, die AnwänderInnen daran zu hindern ihre Daten in andere Formate zu exportieren um diese an die eigene Software zu binden. (“Vendor Lock-In”)

nicht nach Marketing Aspekten designed Kommerzielle Software, wird sehr oft nach Marketeing Aspekten entwickelt. Es soll ein guter erster Eindruck entstehen damit die Software gekauft wird. Das danach ist dort oft nicht so wichtig.

keine Backdoors Da der Source Code offen ist und viele EntwicklerInnen an der Software arbeiten ist es viel schwieriger für Geheimdienste Überwachungs-Hintertüren in Freie Software einzuschleusen.

bessere Qualität und Stabilität Obwohl es natürlich große Unterschiede in der Qualität von Software sowohl im kommerziellen als auch im freien Bereich gibt: generell ist Freie Software oft deutlich besser, schlanker, stabiler und sicherer. Zu Zeiten als Windows 3.1 AnwänderInnen noch mit täglichen oder stündlichen Abstürzen rechnen mussten, hatten Linux AnwenderInnen schon ein System das wochenlang ohne Probleme gelaufen ist.

software leicht zu installieren zumindest die (zigtausenden) Programme die Teil einer Linux Distribution sind, sind sehr leicht (Wie aus einem App-Store) zu installieren.

an eigene Bedürfnisse anpassbar . Notfalls mit Änderungen am Source-Code, sehr oft aber mit einfachen Skripten.

keine Abhängigkeit von Konzernen . Wer kauft die Firma als nächsts auf? Für uns Linux user nicht wirklich relevant.

Hilfe über Community Da das Programm selbst frei ist, sind auch viele AnwänderInnen eher bereit, gratis Hilfestellung für andere BenutzerInnen zu geben. Bei kommerziellen Programmen ist es eher so, dass die UserInnen denken: “Das soll doch der Hersteller machen.”

Gibt es auch Nachteile? Ja: Manchmal ist freie Software nicht so glattpoliert wie manche kommerzielle Produkte. Für manche Aufgaben gibt es auch fast nur kommerzielle Software.

3 Lizenzen und die Politik Freier Software (10 Min)

3.1 Die 4 Freiheiten der GPL

Die GPL “General Public Licence” (Oft auch GNU Licence genannt, da von Stallman für die GNU Projekt entworfen) garantiert 4 Freiheiten:

run for any purpose Niemand darf einschränken wofür man/frau die Software verwenden will.

study and change Der Source Code muss dem Programm beigelegt werden oder einfach verfügbar gemacht werden damit Menschen aus dem Programm lernen können und sie es für sich verändern und erweitern können.

redistribute - help others Niemand darf die Weitergabe des Programmes und des Source Codes einschränken.

redistribute modified versions Niemand darf die Weitergabe von modifizierten Versionen des Programms einschränken. Einzige Einschränkung: Die modifizierten Versionen müssen Ebenfalls unter GPL verfügbar sein.

Vorteil der GPL ist, dass der Code der von der Community geschaffen wurde nicht von kommerziellen Konzernen missbraucht werden kann: Diese dürfen die Software zwar auch, wie alle Anderen, verwenden, müssen aber, wenn sie diese in Produkte einbauen, den Source Code auch unter GPL wieder weiter geben. Oft ist es für Firmen Sinnvoll das zu tun und damit vergrößert sich der Pool freier Software. (Die GPL ist “viral”).

3.2 Andere Lizenzen; BSD, Creative Commons

Die wichtigste anderen freie Lizenz ist die BSD (bzw. BSD artige Lizenzen. z.b. MIT). Sie erlaubt mehr oder weniger “alles”. Allerdings auch: die weiterverwendung des Source Codes in kommerziellen Produkten bei denen der Code **nicht** weiter gegeben wird.

Firmen können BSD Lizenzierte Freie Software daher leicht “stehlen”. Microsoft und Apple haben sehr viel BSD lizenzierten Code verwendet.

Die von Lawrence Lessig entwickelten CC (“Creative Commons”) Linzenzen haben die GNU und die BSD lizenzen als Vorbild. Die Vielfalt der Optionen bei CC bringt aber auch Probleme mit sich.

3.3 Die Politik Freier Software, Keimformen

3.3.1 Keimform

Der Erfolg freier Software in unserer Welt ist ein Beleg dafür, dass diese mit dem Kapitalismus bis zu einem gewissen Grad kompatibel ist. Firmen haben ein Interesse daran. Auch Firmen wie Google oder Apple wollen nicht von Microsoft abhängig sein. Firmen schätzen alles was Kosten senkt und wenn Freie Software das kann dann wird es auch eingesetzt.

Andererseits ist verwirklichte freie Software doch vieles von dem was wir als Kommunismus verstehen. Das Eigentum an Software wird, unter Ausnutzung des Copyrights, aufgehoben und die Freiheit der Software wird damit geschützt. In Freie Software sehen wir die freie Kooperation freier Individuen - soweit diese unter den gegebenen Umständen möglich ist.

Man spricht daher auch von einer "Keimform". Freie Software zielt in Richtung "Überwindung des Kapitalismus", kann aber mit in ihm wachsen.

Ich denke das sollte für uns politische AktivistInnen durchaus Beispielcharakter haben: Wir müssen unsere Aktivitäten so gestalten, dass sie im hier und jetzt wachsen können und gleichzeitig auf die Veränderung eben dieser herrschenden Verhältnisse abzielen.

Zwei weitere Prinzipien aus der Produktion Freier Software die, meiner Meinung nach, für uns Relevant sind: "Forks" und "Show Running Code"

3.3.2 Fork

Ist jemand mit der Richtung in die ein Projekt sich entwickelt nicht einverstanden, so kann er oder sie das Projekt jederzeit "forken". D.h. man/frau macht Entwicklungen in eine andere Richtung. Wenn der Source Code dann auseinander läuft ist das zwar Ineffizient (und somit die Motivation hoch, einen Fork zu vermeiden und doch zusammen an einem Code-Strang zu Arbeiten hoch) aber immer eine Option. Die Freiheit getrennte Wege zu gehen besteht immer und wird ab und zu auch in Anspruch genommen.

3.3.3 Show Running Code

Oft meinen Menschen sie müssten mit guten Anregungen helfen. Wer aber wirklich etwas bewegen will sollte doch mit machen. Linus Torvalds, Ursprünglicher Entwickler von Linux, antwortet mit dem berühmten "Show running Code" einem anderen Entwickler der meinte man solle bestimmte Dinge ganz anders machen.

Hier also für uns als AktivistInnen: Wer es anders will muss auch mitmachen.

Einige dieser Überlegungen sind in den Artikel

<http://qummunismus.at/p/article120.html> eingeflossen.

Mehr Überlegungen zur politischen Relevanz auch auf: http://mond.at/texte/soak05_freie_software.html

4 Commandozeile (CLI) versus Graphische Oberfläche (GUI) (10 Min)

Ein Linux kann heute durchaus gut ohne Kommandozeile benutzt werden. Für fast alles gibt es gute grafische Programme. Dennoch lohnt es sich auch in die Kommandozeile ("Command Line Interface (CLI)") einzusteigen. Eine gut gestaltete CLI hat gegenüber einer grafischen Oberfläche durchaus viele Vorteile:

automatisierbar die Befehle die man/frau händisch eintippt können sehr leicht auch in automatisch ablaufende Scripts eingebaut werden. Wenn z.B. eine Aktion jeden Tag zur selben Uhrzeit ausgeführt werden muss oder eine Aktion auf tausende oder millionen Dateien angewendet werden muss und man nicht alles händisch anklicken will.

leichter dokumentierbar, nachvollziehbar wer schon einmal eine Anleitung für ein grafisches Programm durchgelesen hat: Das ist schwierig zu erstellen und schwierig zu erklären. Klicken sie auf das dritte Tab im Fenster und wählen sie den rot unterlegten Menüpunkt, etc. Dagegen ist es sehr einfach aufzuschreiben welche Befehle eingetippt werden müssen.

weniger Ressourcen Aufwand Eine Grafische Oberfläche benötigt relativ viele Ressourcen (Speicherplatz, CPU Leistung). Eine CLI kann auch auf extrem schwachen Geräten (z.B. Raspberry Pi) genutzt werden.

fernzugriff leicht möglich auch die Benutzung über langsame Internetverbindungen hinweg ist mit CLI sehr bequem möglich.

Ausdrucksstärke . Hast du schon mal versucht dich alleine mit Gestik mit denen Mitmenschen zu verständigen. Die gesprochen und die geschriebene Sprache sind da deutlich Stärker in ihrer Ausdruckskraft. Ähnlich kann auch mit kurzen Befehlen viel mehr ausgedrückt werden wie mit langem herumgeklicke.

Hilfe Eine gute CLI bietet i.a. auch Hilfestellung an. Da die schriftlichen Befehle leichter dokumentiertbar sind, ist diese sogar oft besser als bei grafischen Programmen.

Natürlich hat auch eine GUI ihre Vorteile - insbesondere dort wo auch die Problemstellungen grafischer Natur sind: z.B. ein Zeichenprogramm.

5 Andere Betriebssysteme und gemeinsame Nutzung mehrere Betriebssysteme (10 Min)

5.1 Einordnung (5 Min)

Der Kern von Linux ist der "Kernel". Der wird z.b. auch in **Android** (Googles Betriebssystem für Handys) verwendet. Allerdings hat Android dort nur Google-eigene Programme laufen. Viele davon ohne Source-Code, auch wenn Teile von Android auch unter einer relativ freien Lizenz verfügbar sind.

Der Linux Kernel und zum Teil auch Linux Programme werden in vielen **Appliances** (z.B. WLAN-Router, Alarmanlagen, NAS-Storage Server, etc, etc) eingesetzt. Dort hat man als normaler Anwender meist keinen direkten Zugang zum System sondern nutzt das ganze über ein Webinterface.

Üblicherweise meint man mit Linux aber ein mehr oder weniger vollständiges Linux (**GNU/Linux**) mit allen wichtigen Systemprogrammen. Man kann diese Programme auch immer selbst "compilieren" (d.g. vom Source Code in ein Programm umwandeln). Das ist jedoch aufwendig. Daher nimmt man i.a. fertige Distributionen in denn schon alle wichtigen Programme enthalten sind.

Mac OS/X ist ein kommerzielles Unix von Apple (das aus einem freien BSD Unix heraus entwickelt wurde aber jetzt nicht mehr frei ist). Wer sich auf der Linux Kommandozeile zuhause fühlt wird sich auch auf einem Mac mehr oder weniger zurecht finden. Die grafische Oberfläche ist aber nicht direkt mit der Unix X Oberfläche kompatibel. Viele freie Software und viele Linux Programme wurden auf Mac-OS/X portiert und können dort verwendet werden.

Abwohl Apple enorm von freier Software profitiert hat (Ohne den BSD Kernel wäre ihr hoffnungslos veraltetes und instabiles Mac-OS9 wohl das Ende von Apple am Desktop gewesen) ist Apple der Idee von Freier Software eher feindlich eingestellt.

Freie Programme können natürlich relativ leicht auf andere Systeme portiert werden. Viel der nützlichen Linux Programme wurden daher auch auf Microsoft-Windows portiert. Gimp, Inkscape, Blender, etc. können alle auch dort, mehr oder weniger gut, verwendet werden.

5.2 Windows und Linux gleichzeitig nützen (10 Min)

Wer längere Zeit mit Linux arbeitet will i.a. nicht mehr zurück. Gerade für den Umstieg am Anfang ist aber eine gleichzeitige Nutzung oft hilfreich. Am Ende ist es aber immer besser zu versuchen die Dinge die früher mit kommerziellen Programmen erledigt wurden mit Freien zu versuchen.

Dabei gibt es folgende Möglichkeiten:

dual boot Beide Systeme sind (entweder auf der selben, geteilten Festplatte oder auf verschiedenen Festplatten) am Computer installiert. Beim Booten wählt man mit einem Menü aus welches der Systeme gestartet werden soll. Die Nutzung ist zwar damit nicht ganz "gleichzeitig". Im Grunde aber die sauberste Variante. Jede der Systeme läuft ganz "native" und man hat am wenigsten Probleme mit Inkompatibilitäten.

virtualisierung Mit Virtualisierung wird ein kompletter eigener Computer "simuliert". D.h. Das System "denkt" es läuft auf einer eigenen Maschine, ist aber (wie bei "Matrix") in einer Simulation eines Computers gefangen. Wobei i.a. nur das originale Betriebssystem direkten Zugriff auf die Hardware des Computers hat (Netzwerk, Festplatte, USB, Grafik) und das simulierte System nur eine simulierte Hardware sieht. (Virtualisierungen gibt es auch für Mac OS/X).

Dabei gibt es zwei grundsätzlich verschiedene Möglichkeiten:

1. Mittels KVM im Linux einen Windows Computer simulieren. Das ist i.a. die bessere Variante: Damit ist der größte Schritt für den Umstieg auf Linux bereits vollzogen.
2. Für Virtualisierung gibt es im Windows mehrere Lösungen: das freie QEMU ist frei aber nicht all zu schnell. VMware, VirtualBox, Hyper-V sind kommerzielle Virtualisierungen auf Windows in denen man z.b. einen virtuellen Linux Rechner betreiben kann.

Freie Programme unter Windows . Da Frei Software leicht zu protieren ist, gibt es viele Programme die ursprünglich für Linux entwickelt wurden auch im Windows. Gimp, Inkscape, Libre-Office, etc. Als Vorbereitung für einen später geplanten Umstieg ist das ein guter erster Schritt.

subsystem, API-emulation in Windows Das neue Windows 10 bietet eine Linux-Subsystem an. Die Programme laufen direkt im Windows benutzen aber API Schnittstellen die denen von Linux ähneln. (Ähnlich auch das freie CYGNUS Projekt, das versucht GNU Tools unter Windows zu betreiben). Das ganze ist nützlich wenn man aus irgend einem Grunde Windows benutzen muss, aber doch auch Linux und Unix Programme verwenden will.

wine Das Pendant dazu aus der anderen Richtung ist das **wine** Projekt, das versucht die Windows-API auch unter Linux zu Verfügung zu stellen. Das Windows **kein** freies System ist, und Microsoft diese Projekt natürlich in keinster Weise unterstützt, ist die Aufgabe schwierig. Wine ist daher alles andere als perfekt, hat aber inzwischen schon enorm viel geleistet. Viele Windows-Programme können so mehr oder weniger gut im Linux betrieben werden.

remote arbeiten: von Windows auf Linux und umkehrt wenn man 2 Computer hat dann kann der eine auf Linux und der andere Windows laufen und man kann übers Netzwerk Programme von einer Seite auf der anderen Seite benutzen:

1. von Windows mit putty und SSH die Comandline im Linux benutzen. Braucht man nur die Linux Commandline, dann kann man/frau mit dem freien und sehr guten **Putty** auf ein entferntes Linux (auch am anderen Ende der Welt) einloggen und dort arbeiten.
2. Die grafische Obefläche von Linux (X11) kann sehr leicht auf ein Windows umgeleitet werden. Dazu benötigt man eine X11 emulation. z.b. Xming (frei) oder Exceed (teuer, kommerziell). Diese arbeiten auch gut mit putty zusammen, so dass die Verbindung durch eine Putty-SSH Verbindung getunnelt werden kann. Zuerst Xming starten und dann im Putty auswählen, dass die X11 Verbindung weitergeleitet werden soll.
3. mit dem **VNC** Protokoll kann sowohl eine Windows Oberfläche von Linux aus genutzt werden, als auch umgekehrt.
4. mit rdesktop und remmina kann man die in Windows eingebaute remote-Funktionalität (RDP) nutzen.

Die remote Funktionen sind natürlich auch für den Zugriff auf virtualisierte Systeme nützlich.

live CD siehe Unten. Zum “kurz einmal testen” sind natürlich Live-CD bzw Live-USB Sticks nützlich. Man kann in ein komplettes Linux booten ohne dies auf der Festplatte instalieren zu müssen.

6 Unix/Linux Stammbaum: Die verschiedenen Distributionen (10 Min)

Neben Linux gibt es auch heute immer noch kommerzielle Unixe: z.B.: Solaris (oracle) oder IBMs AIX. Üblicherweise werden auch auf kommerziellen Unixen heutzutage viele GNU Tools installiert. Man/frau fühlt sich dort auch “zuhausse”.

Weiters gibt es viele freie, von BSD abstammende Unixe. z.B. FreeBSD oder OpenBSD. Im vergleich zu Linux führen diese aber eher ein Nieschendasein.

6.1 Distributionen

Da Linux frei ist kann jede/r kommen und es sich so einrichten und zusammenstellen wie er/sie möchte. Das ist viel arbeit aber dennoch gibt es verschiedene Zusammenstellungen (Distributionen). Üblicherweise mit verschiedenen Schwerpunkten und Zielsetzungen.

Eines der wichtigeren Entscheidungskriterien ist, welches Packetmanagment System verwendet wird. Es gibt Debian (.deb) und RedHat (.rpm).

Manche hersteller packen zu den freien Linux Programmen auch noch kommerzielle Programme dazu und/oder bieten kommerziellen Support an.

Hier eine unvollständige Auflistung wichtiger Distributionen

slackware - die erste Linux distribution. Heute kaum noch verwendet

debian - die wichtigste freie Distribution. In der Kern-Distribution sind **nur freie** (im Sinne der DFSG - siehe oben) Programme enthalten. Man kann aber auch “non-free” dazu nehmen. Debian hat ein sehr grosses Repertoire an Programmen und legt Wert auf einen möglichst einfachen und reibungslosen Update-prozess. Alle 2 bis 3 Jahre gibt es eine neue “Major-Release” mit neuen Programmversionen. Dazwischen i.a. nur Security-Updates. D.h. das System ist sehr stabil und bleibt wie es ist.

ubuntu - kommerzielle Distribution. Für den Privatgebrauch Gratis. Basiert auf Debian. Gut für Einsteiger. Es gibt LTS (“Long-Term Stable”) Versionen die ca. 4 Jahre lang gleich bleiben, aber security updates erhalten.

mint - auf debian und ubuntu aufbauende Distribution.

RedHat - kommerzielle Distribution, von IBM gekauft. Wird gerne im Firmenumfeld für Serversysteme genutzt (konservative Update-Politik, guter kommerzieller Support).

Fedora - gratis version von RedHat in dem die allerneusten Features ausprobiert werden können.

CentOS - RedHat clone, Frei.

gentoo - Alle Pakete werden i.a. immer direkt von Source erzeugt. (nicht wirklich für Einsteiger).

Was ist die besten Distribution für EinsteigerInnen. Ich empfehle Debian oder Ubuntu. Persönlich bevorzuge ich Debian.

6.2 Live CDs

Live-CDs sind CDs oder USB-Sticks, die direkt gestartet werden können, **ohne** dass dazu ein System installiert werden muss. Das ist praktisch fürs Ausprobieren aber auch nützlich für trouble-shooting (Wenn z.B. der Computer nicht mehr startet und man herausfinden will was das Problem ist), aber auch für Sicherheitsrelevante Operationen.

Hier 3 wichtige Live Systeme:

Knoppix Der Klassiker von Klaus Knopper. Basierend auf Debian (wobei meist auf die experimentelle unstable oder testing-Zweig gebaut wird). Fokus ist: “Linux ausprobieren”.

Grml Ideal zum Trouble-Shooting. Das Schweizermesser für den Sysadmin.

Tails Fokus: Security und Verschlüsselung für Under-Cover Agenten. Die von Edward Snowden promotete Live-CD.