

Linux Installation

im Rahmen von inf.misc am 12. Dezember 2001
christina.zeeh@studi.informatik.uni-stuttgart.de

Inhalt:

- Crash Kurs Grundlagen
- Distributionen
- Partitionierung
- Installation
- Booten
- Konfiguration
- Paketmanagement
- Hilfe & Dokumentation

Disclaimer: Ich bin für nichts verantwortlich (inklusive Frust), was aus der Lektüre dieser Informationen resultieren könnte. Oder was passieren könnte, wenn man diese Informationen nicht liest. Die meisten Angaben beziehen sich auf Red Hat Linux.

Crash Kurs Grundlagen

Linux ist ein **Mehrbenutzerbetriebssystem**. Der Account mit den meisten Rechten ist der Superuser Account „root“. Bei der Installation wird meist auch das Rootpasswort gesetzt. Auch wenn man selbst als einziger Benutzer auf dem System arbeiten wird, empfiehlt es sich, zusätzlich einen Benutzeraccount anzulegen und diesen für die normale Arbeit zu verwenden, da das Ausführen von Programmen als Root immer auch ein Sicherheitsrisiko in sich birgt. Als Root kann man arbeiten, indem man sich entweder direkt mit dem Benutzernamen „root“ und dem bei der Installation gewählten Passwort anmeldet, oder indem man in einer Shell den Befehl `su -` eintippt; man erhält dann eine sogenannte Rootshell. Hat man mehrere Benutzer auf dem System, die auch (teilweise) Administrationsaufgaben durchführen sollen, empfiehlt sich das Programm `sudo`. Einem Befehl vorangestellt, erlaubt es dem Benutzer, diesen Befehl als Root auszuführen. Er wird zur Sicherheit nach seinem eigenen Passwort (*nicht* nach dem Rootpasswort) gefragt. In der Datei `/etc/sudoers` wird festgelegt, welche Benutzer `sudo` verwenden können und welche Befehle sie damit ausführen können. Alles was über `sudo` getan wird, wird mitgeloggt und kann vom Systemadministrator nachvollzogen werden. `Sudo` empfiehlt sich auch als Schutz vor sich selbst, da man beim Ausführen die freundliche Aufforderung „think before you type“ erhält und ggf. vielleicht nochmal über das, was man gerade tut, nachdenkt, und so Fehler vermeiden kann (bzw. wenn der Fehler schon passiert ist notfalls im Log nachschauen, was für einen Befehl man da eigentlich eingetippt hat).

```
=> man sudo
=> man su
=> man useradd
=> man usermod
=> man userdel
```

Auch **Dateien und Verzeichnisse** sind von diesem Multiuserkonzept betroffen. Eine Datei (im folgenden ist damit auch ein Verzeichnis gemeint), gehört einem Benutzer und

ist einer Gruppe zugeordnet. Der Benutzer (und zusätzlich auch Root) kann entscheiden, wer Zugriff auf eine Datei hat. Dies kann nach Rechten für den Benutzer, die Gruppe und alle anderen Benutzer differenziert werden. Es gibt drei verschiedene Typen von Zugriffsrechten: lesen (r), schreiben (w) und ausführen (x). Während die Semantik von lesen und schreiben offensichtlich ist, bedarf das Ausführrecht einer kurzen Erklärung: Bei Dateien bedeutet dieses Recht, dass der Benutzer eine Datei ausführen kann. Bei Verzeichnissen jedoch bedeutet dieses Recht, dass der Benutzer in dieses Verzeichnis wechseln kann. Um irgendetwas mit einem Verzeichnis oder auch seinen Unterverzeichnissen anfangen zu können, muss das Ausführrecht vorhanden sein.

=> man chown
=> man chgrp

/	/ ist das Wurzelverzeichnis unterhalb dessen sich alle andern Verzeichnisse befinden
boot	
dev	/dev enthält die Gerätedateien (z.B. /dev/hda für die erste IDE Festplatte)
bin	
etc	/bin enthält ausführbare Dateien
home	
mustermx	/etc enthält Konfigurationsdateien u.ä.
mnt	/home enthält die Homeverzeichnisse der Benutzer, hier z.B. das von Max Mustermann
cdrom	
floppy	/mnt enthält nicht-permanent gemountete Datenträger, z.B. ein CD-ROM oder eine Diskette.
usr	
bin	
doc	/usr enthält in seinen Unterverzeichnissen Programme, Dokumentation etc.
local	
sbin	/root ist das Homeverzeichnis des Superusers
root	
sbin	/sbin enthält Programme zur Systemadministration (nur für Root ausführbar)
tmp	
var	/tmp enthält temporäre Daten
	/var enthält variable Daten, z.B. Logfiles oder Spools für Drucker

=> man chmod

Die **Verzeichnisstruktur** eines Linux Systems unterscheidet sich stark von der z.B. eines Windows Systems. Sie ähnelt der Verzeichnisstruktur anderer Unix Systeme. Das Trennzeichen in Pfadangaben ist der /, nicht der \ wie in Windows.

Alles befindet sich in einem einzigen Verzeichnisbaum, ein CD-ROM Laufwerk z.B. wird in den Verzeichnisbaum eingebunden und ist dann unter /mnt/cdrom verfügbar. Genauso kann der Verzeichnisbaum über mehrere Festplatten verteilt werden, so

können z.B. die Homeverzeichnisse der Benutzer (/home) auf der einen Platte und die restlichen Daten (/) auf der anderen liegen. Der Benutzer bekommt von all dem nichts mit.

Mit dem Befehl `mount` kann man Datenträger in den Verzeichnisbaum einbinden. Dazu muß man normalerweise Root sein. In `/etc/fstab` werden Mountpoints aufgeführt, die permanent gemountet sind (z.B. `/` oder `/home`). Außerdem können dort Mounts festgelegt werden, die von jedem Benutzer ausgeführt werden können (z.B. Diskette oder CD-ROM)

```
=> man mount
=> man umount
=> man fstab
```

Dateiendungen haben unter Linux aus Betriebssystemsicht keine Relevanz (d.h. das Betriebssystem bedient sich nicht der Dateiendung, um herauszufinden, mit welcher Anwendung eine Datei zu öffnen ist). Aus Benutzersicht sind Dateiendungen aber trotzdem ganz praktisch. Um herauszufinden, von welchem Typ eine Datei ist, dient der Befehl `file`.

```
=> man file
```

Distributionen

Linux kommt im allgemeinen in sogenannten Distributionen. Jemand nimmt den Kernel (das eigentliche Betriebssystem) und zahlreiche System- und Anwendungsprogramme und packt sie zu einem (hoffentlich) stimmigen System mit (hoffentlich) benutzerfreundlicher Installationsroutine zusammen. Viele Distributionen werden im Karton, mit Handbüchern und CDs zum Verkauf angeboten. Man aber kann eigentlich alle auch kostenlos im Internet herunterladen, wenn man bereit ist, auf Zugaben wie Handbücher, Support und kommerzielle Software zu verzichten. Bekannte Distributionen sind RedHat, SuSE und Debian, daneben gibt es hunderte weniger bekannte Distributionen, sowie spezialisierte Distributionen wie YellowDog (PPC Spezialist) und das Linux Router Project.

Partitionierung

Linux installiert sich am bequemsten auf einer völlig leeren Festplatte. Ist die Festplatte, auf die Linux installiert werden soll, nicht völlig leer, sollte man vorher unbedingt ein Backup anlegen. Partitionieren verursacht meist vollständigen Datenverlust, allerdings ist es in einigen Fällen auch möglich, ohne Datenverlust zu partitionieren, wegen der hohen Fehlergefahr ist ein Backup jedoch immer sinnvoll.

Windows in primärer Partition, die die gesamte Platte einnimmt:

Wenn noch freier Plattenplatz verfügbar ist, kann die primäre Partition nach Defragmentierung mit einem Utility wie FIPS verkleinert werden und dann auf den freien Platz Linux in eine weitere primäre oder auch in eine erweiterte Partition installiert werden.

Windows in primärer Partition, es ist noch unpartitionierter Platz vorhanden:

Linux kann einfach in eine weitere Partition auf dem unpartitionierten Platz installiert werden.

Der Linux-Teil muss mindestens eine Partition für das Wurzelverzeichnis / und eine Swap-Partition umfassen. Für die Swap Partition ist eine Größe, die etwa dem doppelten Hauptspeicher entspricht, empfehlenswert. Alles in allem sollte man für eine Linux Installation mit graphischer Oberfläche und vielen Anwendungsprogrammen mindestens 1-2 GB Plattenplatz einplanen; für Router o.ä. kommt man jedoch mit sehr viel weniger (ggf. lediglich eine einfache Diskette) aus.

Empfehlenswert ist es, eine separate Partition für /home anzulegen, da man diese dann ggf. bei einer Neuinstallation einfach belassen kann und somit seine Benutzerdaten nicht verliert. Je nach Geschmack kann man auch jeden anderen Teil des Verzeichnisbaums physisch auf unterschiedlichen Partitionen und sogar Datenträgern anlegen lassen. Der Vorteil davon ist, dass man ein Betriebssystem auf mehrere kleine Platten verteilen kann, die für sich allein möglicherweise zu klein wären. Wenn man auf einer großen Platte jedoch viele kleine Partitionen anlegt, besteht die Gefahr, den späteren Platzbedarf zu unterschätzen und dann z.B. auf der Home-Partition keinen Platz mehr zu haben, während auf /usr noch viel Platz frei ist.

=> man fdisk

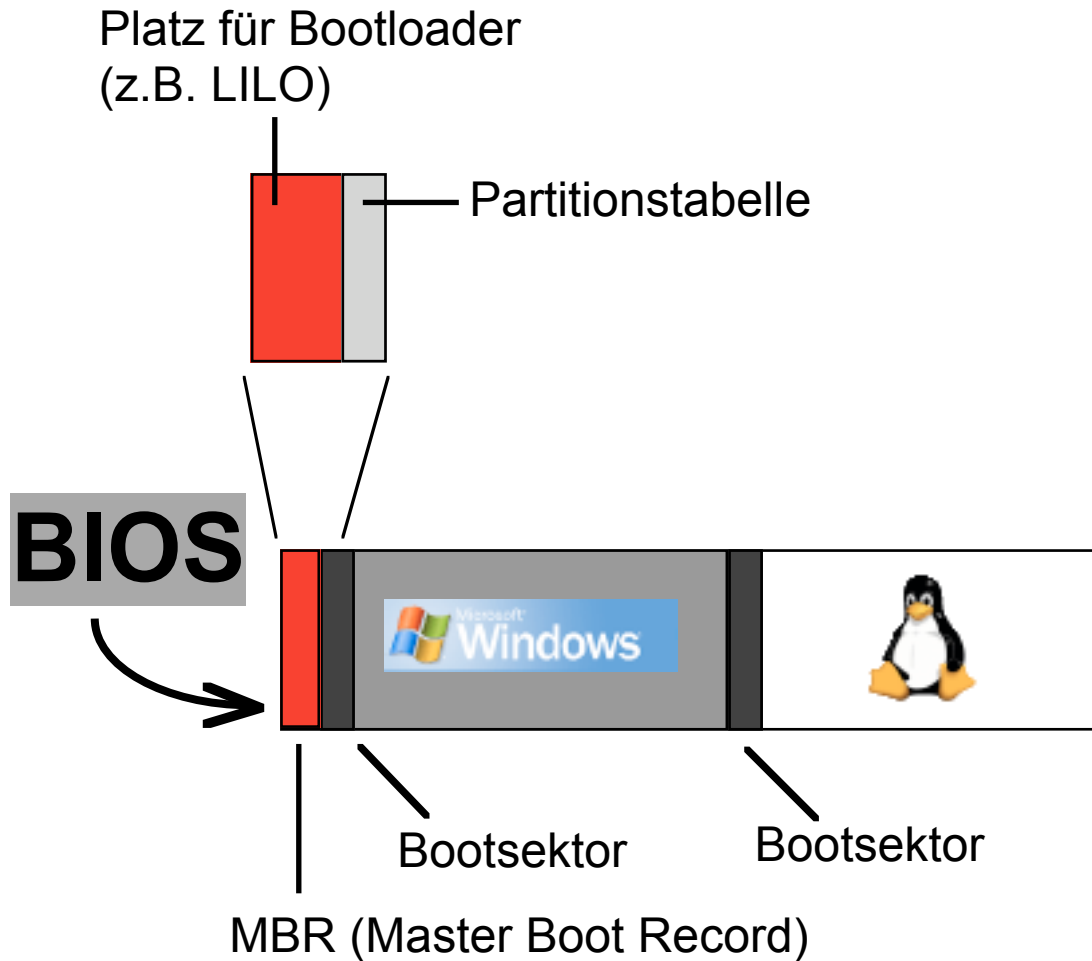
Installation

Wenn man Linux käuflich erwirbt wird man es meist komplett von den mitgelieferten CD-ROMs installieren. Wenn kein bootbares CD-Laufwerk vorhanden ist, kann man ggf. mit einer Bootdiskette booten und dann mit der CD weiterinstallieren oder das Programm `loadlin` unter DOS verwenden. Auch wenn man sich CD-Images aus dem Netz herunterlädt und diese brennt, wird man so verfahren.

Man kann Linux alternativ auch aus dem Netz installieren. Dies macht allerdings nur mit einer günstigen und schnellen Netzanbindung Sinn. In diesem Fall bootet man von Diskette oder CD-ROM und wählt dann im Installationsprogramm als Installationsmethode `http` oder `ftp` aus. Wenn man im lokalen Netz eine Kopie der gewünschten Distribution hat, kann man ggf. auch per NFS installieren. Die Images für die Boot-Datenträger kann man sich von den entsprechenden FTP-Servern der Distributionen herunterladen und dann - im Fall der Diskette - mit dem DOS-Programm `rawwrite` schreiben.

Bei der Installation wird man im allgemeinen aufgefordert, einen **Installationstyp** auszuwählen. Im allgemeinen gibt es Optionen wie Server, Workstation, Laptop oder Benutzerdefiniert/Experte. Im allgemeinen empfiehlt sich eine Experteninstallation oder die Workstation bzw. ggf. die Laptop Variante. Server-Installation sollte man nur auswählen wenn man weiss, was man tut - und in diesem Fall wählt man sowieso die Experten-Installation ;-). Der Installationstyp bietet eine grobe Vorauswahl der zu installierenden Software (Pakete). Desweiteren kann meistens nach Gruppen (Textverarbeitungssystem, Spiele, Webserver, Mathematik, usw.) und auch direkt auf Paketebene ausgewählt werden. Zum Anfang empfiehlt es sich, eine Grobauswahl der Gruppen zu treffen, und dann ggf. in der untersten Ebene einzelne Pakete zusätzlich auszuwählen. **Vorsicht** ist bei allen Diensten geboten, die im Netzwerk nach außen sichtbar sind (Server u.ä.). Oft werden diese automatisch gestartet und können eine Sicherheitslücke darstellen. Diese Pakete daher nur auswählen, wenn man weiss was man tut und auch bereit ist, regelmäßig nach Bugs und Updates zu schauen.

Booten



Was passiert beim Booten?

Das BIOS liest den Inhalt des MBRs in den Speicher ein und führt ihn aus. Typischerweise (= Windowsrechner) befindet sich im MBR ein einfacher Bootloader, der nichts anderes tut, als die aktive Partition zu booten (d.h. deren Bootsektor in den Speicher zu laden und auszuführen).

Linux stellt einen eigenen Bootmanager zur Verfügung. Typischerweise ist dies LILO, in letzter Zeit wird auch immer öfter GRUB verwendet.

Der Bootmanager wird entweder in den MBR geschrieben und bietet dann eine Auswahl zwischen Windows und Linux, oder man schreibt ihn in den Bootsektor der Linux-Partition und setzt diese als aktive Partition. Typischerweise wird man aber die MBR Methode wählen. Der Vorteil der Bootsektor-Methode ist allerdings, dass man ungestraft Windows installieren kann. Windows überschreibt nämlich bei der Installation den MBR wieder mit seinem eigenen Bootloader und löscht so einen eventuell dort befindlichen LILO. Dies kann allerdings mit einer Linux-Boot-Diskette/CD und neuschreiben des LILO behoben werden.

Für späteres Kompilieren eines Kernels sollte man sich merken, dass nach der Neukompilierung LILO neugeschrieben werden muss: LILO kann keine Dateisysteme lesen, sondern merkt sich den zu ladenden Kernel anhand seiner Festplattenposition (und dessen Position ändert sich typischerweise bei einer Neukompilierung).

Die Konfiguration von LILO findet sich in `/etc/lilo.conf`. Anhand dieser Konfigurationsdatei kann LILO mit dem Befehl `lilo` (als Root) neugeschrieben werden.

```
=> man lilo
```

Konfiguration

Der **X-Server** (graphische Oberfläche) kann üblicherweise direkt während der Installation konfiguriert werden. Heutzutage werden die gängigen Grafikkarten und Monitore automatisch erkannt, gelegentlich (insbesondere bei älteren oder selteneren Modellen) kann es jedoch Probleme geben. Hilfreich ist es dann, möglichst viele Informationen zu Monitor und Grafikkarte parat zu haben, um die korrekte Kombination manuell einstellen zu können. Und wichtig: unbedingt die eingestellte Konfiguration testen, bevor man dem Installationsprogramm sagt, dass X automatisch gestartet werden soll.

Bei einem laufenden Linux liegt X normalerweise auf Konsole 7 (Ctrl-Alt-F7) und auf den Konsolen 1-6 (Ctrl-Alt-F1 bis F6) sind Textkonsolen verfügbar, die auch dann funktionieren, wenn der X-Server (noch) nicht richtig konfiguriert ist oder nicht gestartet ist.

```
=> man Xconfigurator
```

```
=> man XF86Config
```

```
=> man X
```

Bei der **Netzwerkconfiguration** sind kaum Besonderheiten zu beachten, Linux unterstützt von Haus aus sämtliche gängigen Netzwerkkarten und -protokolle und bringt Möglichkeiten für Routing und Firewalling (mit iptables oder in älteren Versionen ipchains bzw. ipfw) mit.

Für die Konfiguration von Modem, ISDN und DSL sind spezielle Tools - je nach Distribution heißen diese unterschiedlich - verfügbar (kppp, adsl-config, wvdial, rppp, linuxconf, usw.)

```
=> man pppd
```

```
=> man resolv.conf
```

Paketmanagement

Neue Software kann man entweder als Sourcecode oder Binärdateien bekommen, oder als Source- oder Binärpakete. Pakete umfassen alle Dateien, die für ein Programm erforderlich sind, sowie genaue Informationen über die Zielorte der Dateien, die Version und benötigte sowie zur Verfügung gestellte Bibliotheken. Das Paketmanagementsystem verwaltet diese Pakete und stellt sicher, dass alle Abhängigkeiten beachtet werden. So kann Software problemlos installiert und auch wieder vollständig deinstalliert, sowie geupdatet werden.

Red Hat und SuSE sind RPM-basierte Distributionen (**RedHat Package Management**). Debian verwendet für seine Pakete das DEB Format. Beide Systeme haben ihre Vor- und Nachteile, wobei Debian wohl das bessere System zum updaten und zum Auflösen von Paketabhängigkeiten besitzt.

Einige Befehle für das RPM-System (es gibt auch graphische Front-Ends für RPM):

```
rpm -ivh <Datei(en)> installiert das/die entsprechenden Pakete
rpm -Uvh <Datei(en)> updated bereits installierte Pakete
rpm -e <Paket(e)>      löscht das/die installierte(n) Paket(e)
rpm -q <Paket>        gibt Informationen über das Paket aus (Optionen s. Manpage)
```

Wie oben erwähnt, kann man Software auch im Sourcecode beziehen. Diese kommt dann meistens in gezippten TAR-Archiven (Endung .tar.gz oder .tgz). Folgendes Vorgehen kompiliert und installiert die Software:

<code>tar xvfz <Datei></code>	Archiv auspacken
<code>cd <Archivverzeichnis></code>	
<code>./configure</code>	Configure-Skript ausführen *
<code>make</code>	kompilieren
<code>make install</code>	installieren (als Root)

* dem Configure-Skript können zahlreiche Optionen übergeben werden. Interessant ist die Option `--prefix="<Pfad>"`. Damit kann ein alternatives Installationsverzeichnis angegeben werden, was besonders interessant ist, wenn man auf einem Rechner keine Schreibrechte für das voreingestellte Installationsverzeichnis hat. In diesem Fall muss `make install` nicht als Root ausgeführt werden.

Der Nachteil von selbstkompiliertem Sourcecode (nicht verwechseln mit Sourcepaketen) ist, dass diese Software nicht dem Paketmanagementsystem untersteht. Wenn man das ausgepackte Archivverzeichnis nach der Installation belässt, ist es jedoch möglich, die Software automatisch mit dem Befehl `make uninstall` zu deinstallieren.

Hilfe & Dokumentation

Manpages:

Mit dem Befehl `man <Befehl>` lässt sich die Hilfe zu praktisch jedem installierten Programm/Befehl anzeigen.

HOWTOs und Tutorials:

Auf <http://www.linuxdoc.org> finden sich themenorientierte HOWTOs und Tutorials. Unbedingt lesen!!!

Mailinglisten(-archive), Newsgroups und FAQs:

In Mailinglisten und Newsgroups und insbesondere auch in deren Archiven findet sich eine Fülle konkreter Informationen zu bestimmter Software oder allgemeinen Themen. Auf jeden Fall vorher die Archive und FAQs lesen, bevor man eine Frage auf eine Mailingliste schickt.

IRC:

#linux.de oder #linuxger (Netiquette beachten und vorher die Webseite unter <http://www.linuxger.de> durchlesen)

Linux User Groups:

z.B. in Stuttgart <http://www.lug-s.org>