

Linux Installation

Steve Moser, Markus Ganzenmüller

9. November 2002

Inhaltsverzeichnis

1	SuSe 7.3	2
1.1	Distributionen	3
1.2	Linux-Installation SuSe 7.3	3
1.2.1	Vorgaben	4
1.2.2	Installationsvorbereitung	5
1.2.3	Startvorgang	5
1.2.4	Konfiguration mit YAST2	5
1.2.5	Paketauswahl	6
1.2.6	root, user	6
1.3	Konfiguration	7
1.3.1	Einleitung	7
1.3.2	Konfiguration des Systems mit SaX	7
1.3.3	LILO (LinuxLoader)	11
1.4	Fazit	17

Kapitel 1

SuSe 7.3

1.1 Distributionen

Es gibt viele Linux-Distributionen, die man heutzutage erwerben kann. Linux ist an sich eigentlich nur der Kernel. Dieser stellt die Verbindung zwischen den Programmen und der Hardware dar. Es fehlen allerdings sämtliche Befehle (bzw kleinen Programme), die ein Betriebssystem ausmachen.

Diese Befehle sind erst in einer Linux Distribution zu finden. Außerdem enthalten die Distributionen die Anwendungsprogramme wie z.B. das Satzsystem LaTeX, den C Compiler gcc und andere. Die meisten dieser Programme existieren schon länger als Linux und stammen von der GNU (GNU is not UNIX) Organisation. Es gibt mehrere Distributionen, die von verschiedenen Gruppen und Firmen für bestimmte Anwendergruppen ausgelegt worden sind. So haben sich manche Firmen auf Distributionen speziell für Firmenkunden und andere auf Distributionen für Endanwender oder Entwickler spezialisiert.

Da die meiste Software wie Linux selbst der GNU GPL unterliegt, sind die meisten Distributionen auch kostenlos aus dem Internet beziehbar. Ein Download ist jedoch kaum sinnvoll, da eine Distribution leicht eine Größe von 100 MB und mehr erreicht. Ein Großteil der Anwender installiert Linux von CDROMs, da diese inklusive Handbüchern bereits sehr günstig zu bekommen sind.

Viele Anwender haben jedoch Probleme, bei der großen Auswahl an Distributionen die richtige für sich zu finden. Auch in den Newsgruppen entbrennen regelmäßig Glaubenskriege über die richtige Distribution. Letztendlich ist die Wahl eine Frage des eigenen Geschmacks.

Im Prinzip kann man bei der Wahl der Distribution nicht soviel falsch machen. Persönlich würde ich S.u.S.E., Debian und RedHat empfehlen, da diese leicht zu installieren sind und aufgrund der weiten Verbreitung leicht Hilfe in Newsgruppen zu finden ist.

Für die S.u.S.E. Distribution spricht vor allem die gute Anpassung an die deutschen Bedürfnisse und das umfangreiche Handbuch.

Die Debian Distribution ist vor allem für Fortgeschrittene zu empfehlen, die volle Kontrolle über ihr System wünschen. Vorteilhaft ist auch, daß jeder die Möglichkeit hat, selbst aktiv die Entwicklung dieser Distribution zu unterstützen.

1.2 Linux-Installation SuSe 7.3

Lange Zeit galt Linux als ein sehr schwer zu installierendes Betriebssystem. Man musste Dateien zusammensuchen, die Kernelparameter einstellen und diesen speziell für das verwendete System kompilieren. Mit den neueren Linux-Versionen ist das aber einfacher geworden. Man kann Linux alleine oder parallel zu Windows installieren. Die Hauptarbeit ist dann das richtige Aufteilen der Festplatte.

1.2.1 Vorgaben

Installieren sie eine Linux Distribution und richten es Netzwerkfähig ein.
Zentraleinheit:Desktop mit AMD-K6 233 MHz, 64 MB und:

```
1  * Graphik: ATI 3D RAGE PRO (4 MB)
2  * Netz: 3com 3C509b PnP
3  * Dual IDE
4  * cdrom: Mitsumi FX320S, 32x
5  * hda 128/63/781: 3 GB ST33232A
6          hda1: 100 MB root
7          hda2: 100 MB swap
8          hda3: 100 MB /var
9          hda4 extended:
10         hda5: 100 MB /scratch
11         hda6: 1.5 GB /usr
12         hda7: 500 MB /opt
13         hda8: 500 MB /home
14
15  Peripherie
16
17     * Tastatur: pc101 englisch (US)
18     * Maus: 3 Knopf Mouse Systems (-t msc) ttys0-com1:
19     * Monitor: 64KHz Hyunday HL5864E
20     * IP-Adresse: 141.82.30.98 (minos)
21
22  Netzwerk:
23  Rechnername: <zusammen mit IP-Adresse besorgen!>
24  Domainname: informatik.fh-augsburg.de
25  Echtes Netzwerk, kein DHCP
26  Netzwerk Typ: eth0
27  Netmask: 255.255.240.0
28  Nameserver:
29      141.82.16.200      primestation.rz
30      141.82.16.242      av2.rz
31  Nameserverdomains:
32      informatik.fh-augsburg.de
```

```
33         rz.fh-augsburg.de
34 Hardware: eth0, 3com 3c509
35
```

1.2.2 Installationsvorbereitung

Da heutzutage standardmäßig Betriebssysteme von CD installiert werden muß man im Bios die nötige Bootreihenfolge einstellen. Mit dem zuständigen Betreuer wurde ein root Passwort vereinbart.

1.2.3 Startvorgang

Der PC bootet nun von CD und der Kernel wird geladen. Nach der Auswahl von Installation/System startet ein weiterer Installationsmanager, der Yast2. (Yet another setup tool, zweite version). Der Hauptunterschied zu Yast1 liegt darin dass Yast2 mehr auf grafische Oberfläche und vereinfachte Eingaben (wizard technologie) Wert legt. Der Vorteil liegt darin dass auch Anfänger sehr schnell die Möglichkeit haben einen Einstieg in Linux zu bekommen, ohne dafür mehr ins Detail gehen zu müssen. Der Nachteil ist daß, sehr viel unter der grafischen Oberfläche passiert und man keinen Einblick hat was wo, wie installiert wird. Da laut unseren Vorgaben ein schnelles Installieren nicht möglich ist, haben wir zwar den Yast2 gewählt, aber doch die Feinheiten manuell abgestimmt.

1.2.4 Konfiguration mit YAST2

Yast2 schlägt eine Standardinstallation vor. Da wir aber viel eigenständig konfigurieren möchten, wählen wir den Expertenmodus. Dort partitionieren wir laut Vorgabe und geben die Mountpunkte an. Interessant ist an dieser Stelle, wenn eine Partition, z.B. eine VFAT32 Partition von Windows, eingebunden werden soll, erstellt YAST2 automatisch ein Unterverzeichniss /windows/<buchstabe>. Also wird hier mit 'mkdir' im Hintergrund ein Verzeichnis erstellt und als mountpoint in die /etc/fstab eingetragen. Die Datei /etc/fstab enthält alle Informationen über angeschlossene Geräte und die Art und Weise wie, wann und wer sie mounten darf. Auch wird dort festgelegt, wann eine Partition gemountet wird und in welchem modus. Nachfolgend ein Beispiel für die Datei fstab (file system table):

```
/dev/hda6 / reiserfs defaults 1 2
/dev/cdrecorder /media/cdrecorder auto ro,noauto,user,exec 0 0
/dev/cdrom /media/cdrom auto ro,noauto,user,exec 0 0
```

```

devpts /dev/pts devpts defaults 0 0
/dev/fd0 /media/floppy auto noauto,user,sync 0 0
proc /proc proc defaults 0 0
usbdevfs /proc/bus/usb usbdevfs noauto 0 0
/dev/hda2 /drives/se vfat auto,user 0 0
/dev/hda3 /drives/w2k ntfs ro,auto,user,umask=022 0 0
/dev/hda4 /drives/winxp vfat auto,user 0 0
/dev/hda5 /drives/filme vfat auto,user 0 0
/dev/hdb5 /drives/mp3 ntfs ro,auto,user,umask=022 0 0

```

An erster Stelle steht das jeweilige Gerät, oder die entsprechende Partition die auf verschiedenen Festplatten liegen kann. (z.B. hda -> primary harddiskdrive). In der zweiten Spalte findet man das Verzeichnis, in das die Partition gemountet werden soll. Die 3. Spalte gibt das Dateiformat an. Auto, oder noauto gibt an ob die Partition bei Bootvorgang automatisch gemountet werden soll oder nicht. Gefolgt von der Gruppe die dieses Gerät mounten darf, hier die Gruppe users. Auch wird die Art der Rechtevergabe gewählt.(z.B. exec für execute oder ro für read only.) Die letzten zwei Ziffern geben an, wie mit dem Medium verfahren werden soll.(z.B ob das Medium beim starten auf Konsistenz überprüft werden soll,usw)

1.2.5 Paketauswahl

Da die Festplattengröße sehr mager ausfällt kann keine 'Normalinstallation' (Installation aller Pakete die nötig sind für den KDE Window Manager) vorgenommen werden. Um trotzdem eine Grafische Benutzeroberfläche erhalten zu können wählen wir den Blackboxmanager und den XServer, weiterhin werden die Developerpakete gcc und alle abhängigen Pakete ausgewählt, sowie alle nötigen Netzwerkpakete.

1.2.6 root, user

Nach der Installation vorher genannter Pakete werden mindestens zwei Benutzer angelegt. Zum einen der 'root', der Systemadministrator, der über alle Rechte verfügt und sämtliche System einstellungen verändern kann, und zum anderen ein user (z.B. Steve), der nur über eingeschränkte Rechte verfügt. Dies hat den Sinn dass man als normaler User nicht aus versehen oder absichtlich das System beschädigen kann. "Never change a running system" würden manche Leute dazu, mit Recht, sagen.

1.3 Konfiguration

1.3.1 Einleitung

Die folgenden Kapitel zeigen Ausschnitte aus der Konfiguration, welche unter der grafischen Oberfläche von YAST2 bei der Installation vorgenommen wurden, bzw. Programme und Möglichkeiten zur nachträglichen Manipulation wichtiger Konfigurationsdateien. Dadurch ergibt sich die Möglichkeit, das System den Benutzerwünschen bestmöglich anzupassen. Leider können aus Zeitgründen nur einige wenige Themen abgehandelt werden. Zur weiteren Information sei auf die Internetseiten des Herstellers SuSe unter <http://www.suse.de> verwiesen.

1.3.2 Konfiguration des Systems mit SaX

SaX steht für "SuSE Advanced XF86-configurator" und ist eine Alternative zur herkömmlichen X-Konfiguration.

Mit SaX lässt sich die grafische Oberfläche konfigurieren. Folgende Rechnerdaten sollten vor dem Aufruf von SaX bekannt sein. Monitor (Typ), Tastatur, Maus (+Schnittstelle) und Hersteller und Typ der Grafikkarte.

Ein Wizard leitet den Anwender durch die verschiedenen Registerkarten wo die Gerätedaten erfasst werden. Die Daten der erfassten Einstellungen werden in der Datei `/etc/XF86Config` gespeichert. Zur Konfiguration gibt es auch andere Konfigurationstools wie zum Beispiel "xf86config".

Im folgenden werden Teile aus einer Testdatei aufgeführt und erläutert:

Die Konfigurationsdatei ist in verschiedene 'sections' aufgeteilt. Die erste davon betrifft den zu verwendenden Keyboard-Treiber. Es handelt sich bei der Tastatur um eine Standard-Version mit 102 Tasten und deutschem Tastaturlayout, wobei die Konfigurationsvariante der des `xfree86`-Servers entspricht.

```
1          #-----#
2          # Sample Config #
3          #-----#
4          #
5          # Keyboard Settings
6          #
7          Section "Keyboard"
8              Protocol          "Standard"
9              XkbRules          "xfree86"
10             XkbModel          "pc102"
```

```

11         XkbLayout          "de"
12         XkbVariant         "nodeadkeys"
13     EndSection
14

```

In der 'Pointer' section wird das Mausprotokoll und die Schnittstelle, also die Anschlußvariante für die Maus definiert. In diesem Fall handelt es sich um eine Microsoft Maus, welche unter /dev/cua0 (PS/2) ins Dateisystem eingebunden wird. Ferner wird die Emulation von 3 Mausbuttons unterstützt.

```

15     # Mouse Settings
16     #
17     Section "Pointer"
18         Protocol           "Microsoft"
19         Device              "/dev/cua0"
20         Emulate3Buttons
21     EndSection
22

```

Eine weitere section betrifft den verwendeten Bildschirm. Unter anderem werden hier ein Name für den Monitor, Herstellername und Modellname vergeben. Ferner sind die horizontalen und vertikalen Intervalle der Refreshraten in Hertz/Mhertz festgehalten. Die (VESA-)Modelinezeile bestimmt verschiedene Variablen des Synchronisationspulses der auf die Bildröhre trifft. Normalerweise stehen alle Werte hintereinander in einer Zeile, wurden zu Zwecken der Dokumentation jedoch untereinander geschrieben. Die eigentliche Berechnung der Impulswerte ist ziemlich kompliziert und wurde deshalb weggelassen. Abhängig sind diese Werte von der Auflösung, bzw. der Clockfrequenz und der Impulsverzögerung, welche bei jeder Bildschirmart variiert. Der Vollständigkeit halber sei auf eine gute Dokumentation verwiesen, welche die Berechnung der verschiedenen Werte genauer erläutert:

<http://www.de.daemonnews.org/199901/xpert.html>.

```

23     #
24     # Monitor Section with 640x480 VESA Mode
25     #
26     Section "Monitor"
27         Identifier         "Primary Monitor"
28         VendorName         "Unknown"
29         ModelName          "Unknown"
30         HorizSync          25-40
31         VertRefresh        47-75\

```

```

32          Modeline  "640x480" #Auflösung in Bildpunkten
33          29.00          #Clockfrequenz in Mhz
34          640          #Horizontale Pixelanzahl
35          648          #Startpunkt des Impulses
36          672          #Endpunkt
37          771          #Gesamtwert
38          480          #Vertikale Zeilenanzahl
39          490          #Startpunkt
40          495          #Endpunkt
41          525          #Gesamtwert
42
43          EndSection
44

```

Die manuelle Manipulation der Modelines ist eigentlich nur dann notwendig, wenn die Standard VESA Modelines Probleme mit dem Monitor verursachen oder spezielle Frequenzen (z.B. bei Festfrequenz Monitoren) benötigt werden. Dazu muß Linux im Runlevel 3 gebootet werden, damit die grafische Oberfläche (nachfolgend X) deaktiviert wird. Hierzu muß der folgende Befehl als root ausgeführt werden:

```
init 3
```

Modelines lassen sich mit dem Befehl `xmode` errechnen. Um eine Modeline zu errechnen müssen die Horizontal- und Vertikalauflösung sowie die Bildwiederholrate angegeben werden. Die Syntax für das Kommando lautet:

```
xmode -x -y -r
```

Beispiel: Eine Modeline für 1024*768 bei einer Bildwiederholrate von 70 Hz soll errechnet werden. Daraus ergibt sich mit dem oben genannten Kommando folgender Befehl:

```
xmode -x 1024 -y 768 -r 70
```

Die Ausgabe sieht wie folgt aus:

```
51
64
Modeline "1024x768"71.39 1024 1040 1216 1400 768 768 776 802
```

Die Zeile

```
Modeline "1024x768"71.39 1024 1040 1216 1400 768 768 776 802
```

ist die, die anschliessend in die Datei /etc/X11/XF86Config unter Section 'Modes' eingetragen wird.

Es sei aber eindringlich davor gewarnt, ohne vorherige Kenntnis über die genauen Spezifikationen des Monitors, Modelines selbst einzutragen, da ansonsten der Monitor zerstört werden kann!!!

Die letzte section, die hier kurz erläutert werden soll, befaßt sich mit der Grafikkarte. Im unten stehenden Ausschnitt wurden zufällige Beispielwerte verwendet.

Der vergebene Name für die Grafikkarte lautet 'NVIDIA Geforce2', die Karte wurde von NVIDIA hergestellt und besitzt den sogenannten 'NV20' Chipsatz. Die Option 'usevclk1' bezeichnet die Art des verwendeten Clockchips auf dem Board

```
45          #
46          # Section Device
47          #
48          Section "Device"
49              Identifier      "NVIDIA Geforce2"
50              VendorName     "NVIDIA"
51              BoardName      "NV20"
52              Option         "usevclk1"
53          EndSection
54
```

1.3.3 LILO (LinuxLoader)

In diesem Kapitel sollen die Funktionen des LILO, bzw. des linuxloaders unter die Lupe genommen werden, mit dem sich das fertig installierte System booten läßt. Dazu werden zur Verständlichkeit zuerst einige Details zum Bootprozess eines PCs erläutert. Desweiteren wird davon ausgegangen, daß LILO im Master Boot Record installiert werden soll und sich ein weiteres Betriebssystem auf dem Rechner befindet.

Der Bootvorgang

Nach dem Einschalten des Rechners werden vom BIOS Bildschirm und Tastatur initialisiert, sowie der Hauptspeicher getestet. Bis zu diesem Zeitpunkt existieren noch keine Massenspeichermedien in der 'Welt' des PCs. Nachdem das Rumpfsystem seine 'Innenschau' beendet hat, kann es sich der Erkundung der übrigen Welt widmen. Informationen über das aktuelle Datum, Zeit und eine Auswahl der wichtigsten Peripherie-Geräte werden aus den CMOS-Werten ausgelesen. Da nun die erste Festplatte einschließlich ihrer Geometrie bekannt sein sollte, kann das Laden des Betriebssystems von dort beginnen. Dazu wird von der ersten Festplatte der physikalisch erste Datensektor von 512 kb Größe in den Speicher geladen, und die Programmkontrolle geht auf das kleine Programm (in unserem Fall LILO) zu Beginn dieses Sektors über. Die Abfolge der auf diese Weise ausgeführten Anweisungen bestimmt den weiteren Ablauf des Bootvorgangs. Deshalb wird dieser Bereich der Festplatte auch MBR = Master Boot Record genannt.

Komponenten des LILO

LILO ist ein vielseitiger Bootmanager, welcher beim Start Bootsektoren von Partitionen und den Linux-Kernel laden kann. Ferner ist es möglich, dem Kernel diverse Kommandozeilenparameter zu übergeben. Außerdem lassen sich einzelne LILO-Dienste per Passwort schützen.

Die LILO-Startmaschinerie umfasst folgende Komponenten:

1. Einen LILO-Bootsektor mit einem Anfangsstück das den eigentlichen LILO beim Systemstart aktiviert.
2. Den LILO-Maschinencode. (Standardlocation: */boot/boot.b*)

3. Eine *Map-Datei*, in der LILO bei seiner Installation einträgt, wo die Linuxkernel und sonstige Daten, die er braucht, zu finden sind. (Standardlocation: */boot/map*)
4. Optional: eine *Message-Datei*, deren Inhalt vor der LILO-Bootauswahl als Begrüßung ausgegeben wird. (Standardlocation: */boot/message*)
5. Die verschiedenen Linux-Kernel und Bootsektoren, die LILO zum Starten anbieten soll.

Abhängig von der BIOS-Version des Rechners kann es notwendig sein, alle Bestandteile des LILO und das Kernelimage (*/boot/vmlinuz*) innerhalb der ersten 1024 Zylinder der Festplatte zu legen. Notfalls kann man dies durch eine kleine Extrapartition erreichen, welche unter dem Verzeichnis */boot* eingehängt wird und die komplett innerhalb der ersten 1024 Zylinder liegt. Bei vielen BIOS-Versionen älteren Datums sind nur diese physikalischen Bereiche schon während der Systemstartphase mit den BIOS-Treibern erreichbar. Noch dazu ist man in der Regel auf die ersten beiden Festplatten eingeschränkt.

LILO im MBR

Diese Variante bietet größtmögliche Flexibilität, insbesondere ist dies die einzige Möglichkeit, Linux von der Festplatte aus zu booten, wenn sämtliche Linuxpartitionen auf der zweiten Festplatte liegen, und auf der ersten keine erweiterte Partition verfügbar ist. Eine Veränderung des MBR birgt aber auch bei sachgemäßer Installation gewisse Risiken.

Aufbau der *lilo.conf* und Konfiguration des LILO

Als flexibler Bootmanager bietet LILO zahlreiche Möglichkeiten, seine Konfiguration den erforderlichen Befürfnissen anzupassen. Die wichtigsten Optionen und Anpassungsmöglichkeiten sollen im folgenden erläutert werden. Die Konfiguration von LILO wird in der Datei */etc/lilo.conf* eingetragen. Bei unserer Installation im Praktikum wurde dies von YAST2 durchgeführt.

Vor einer Änderung der Konfigurationsdatei sollte die alte Version gesichert werden, um im Notfall darauf zurückgreifen zu können. Eine Änderung wird erst wirksam, wenn LILO zusammen mit der neuesten Fassung der Konfigurationsdatei neu installiert wird.

Im folgenden Abschnitt soll eine Beispielkonfiguration der Datei `lilo.conf` erläutert werden: Die `/etc/lilo.conf` beginnt mit einem globalen Abschnitt mit allgemeinen Einstellungen, gefolgt von einem oder mehreren Systemabschnitten für einzelne Betriebssysteme, die LILO starten soll. Ein neuer Systemabschnitt wird jeweils eingeleitet durch eine Zeile mit der Option `image` oder `other`. Die Reihenfolge der Einträge spielt insofern eine Rolle, als daß das zuerst aufgeführte image nach einer gewissen Wartezeit (`delay/timeout`) zuerst gebootet wird.

```
1      ### LILO global section
2      boot    =    /dev/hda      # LILO installation
3                                 # target: MBR
4      backup  =    /boot/MBR.hda.021028  # backup
5                                 #file for
6                                 #the old MBR
7      vga     =    normal      # normal text mode
8                                 # (80x25)
9      read-only
10     menu-scheme    =    Wg:kw:Wg:Wg
11     lba32          # Use BIOS to ignore
12                  # 1024 cylinder limit
13     prompt
14     password = dva2t      # password example
15     timeout   = 80      # wait at prompt 8s
16                  # before default os
17                  # is bootet
18
19     message    =    /boot/message # LILOs greeting
20
21     ### LILO Linux section (default)
22     image     =    /boot/vmlinuz      # Default
23     label    =    linux
24     root     =    /dev/hda7      # root partition
25                  # for the kernel
26     initrd   =    /boot/initrd
27
28     ### LILO Linux section (fallback)
29     image     =    /boot/vmlinuz.suse
30     label    =    suse
31     root     =    /dev/hda7
32     initrd   =    /boot/initrd.suse
33     optional
```

```

34
35     ### LILO other system section (DOS/Windows)
36     other  = dev/hda1      # Windows partition
37     label  = windows
38
39     ### LILO memtest section (memtest)
40     image  = boot/memtest.bin
41     label  = memtest86
42

```

Alle Zeilen mit einem # am Anfang sind nur Kommentare. Sie werden von LILO komplett ignoriert und dienen nur zur Verbesserung der Lesbarkeit für den Benutzer.

Erläuterung:

boot=<bootdevice>

Device, auf dessen erstem Sektor der LILO-Bootsektor installiert werden soll. Dies kann eine Partition, eine ganze Festplatte, oder eine Diskette sein. Fehlt diese Angabe, wird LILO auf der gegenwärtigen Linux-Rootpartition installiert.

lba32

Diese Option umgeht die 1024-Zylinder-Grenze von LILO. Dies funktioniert natürlich nur, wenn das BIOS das auch unterstützt.

prompt

Erzwingt das Erscheinen der LILO-Eingabeaufforderung. Die Voreinstellung ist: kein Prompt!. Empfohlen, sobald LILO mehr als nur ein System starten soll. Zusammen damit sollte auch die *timeout*-Option gesetzt werden, damit ein automatischer Reboot möglich ist, wenn keine Eingabe am Prompt erfolgt.

timeout=<zehntelsekunden>

Setzt eine Auszeit für die Eingaben des Benutzers am Prompt und ermöglicht dadurch einen automatischen Reboot, wenn nicht rechtzeitig eine Eingabe erfolgt.

image=<kernelimage>

Hier muß der Name des zu bootenden Kernel-Images stehen. Dies wird in der Regel */boot/vmlinuz* sein.

label=<name>

Innerhalb der Datei */etc/lilo.conf* eindeutiger, aber sonst frei wählbarer Name für das System. Durch Eingabe dieses Namens wird beim Systemstart das zu startende Betriebssystem ausgewählt.

root=<rootdevice>

Damit gibt LILO dem Kernel die Rootpartition des Linuxsystems an.

optional

Sollte */boot/vmlinuz.suse* gelöscht werden, wird bei der LILO-Installation dieser Abschnitt ohne Fehlermeldung übergangen.

other=<partition>

Mit *other* werden dem LILO Startpartitionen anderer Systeme zum Booten bekannt gemacht.

label=<name>

Der frei wählbare Name für dieses System.

backup=<backup-Datei>

Die Datei, in der LILO ein Backup des derzeitigen Bootsektors ablegt, in der er anschließend installiert wird.

Im folgenden sollen noch einige Optionale Argumente erläutert werden, um LILO zu konfigurieren:

loader=<Boot-Loader>

Für das Laden eines fremden Bootsektors baut LILO in seiner Map-Datei einen 'Pseudo'-MBR. Beim Booten startet LILO erst diesen und der 'Pseudo'-MBR dann den fremden Bootsektor. Diese Option gibt die Datei an, aus der der Code für den 'Pseudo'-MBR zu nehmen ist.

Manchmal soll ein Betriebssystem, das von der ersten Platte gebootet werden will (z.B. DOS) dennoch mit LILO von einer anderen Platte aus gestartet werden. Die Zusatzoptionen **map-drive=<Nummer>** und **to=<Nummer>** gestatten es, diese beiden Platten anhand ihrer BIOS-Gerätenummern zu vertauschen.

table=<ptabelle>

<ptabelle> muß das Quell-Device für die Partitionstabelle angeben, die in den Pseudo-MBR soll. (Standard: */dev/hda* oder */dev/sda*)

disk=<Geräte-datei>
bios=<BIOS-Geräte-nummer>
cylinders=<Anzahl>
heads=<Anzahl>
sectors=<Anzahl>

Hier kann LILO für einzelne Festplatten direkt vorgeschrieben werden, welche BIOS-Geräte-nummer und Geometrie er zur Adressierung von Sektoren dieser Platte verwenden soll. Das wichtigste Anwendungsgebiet betrifft Misch-Systeme, bei denen SCSI- **und** IDE-Festplatten benutzt werden.

linear

Die Angabe dieser Option bewirkt, daß bei der Installation von LILO sämtliche Referenzen auf Plattensektoren als logische, anstelle physikalischer Adressen, abgelegt werden, sodaß sie unabhängig von der Festplattengeometrie werden. Diese Option ist für den Fall gedacht, daß bei manchen Plattencontrollern das BIOS beim Systemstart eine andere Geometrie erkennt, als das laufende System. Die linear-Option befreit aber nicht von der 1024-Zylinder-Grenze, die durch die BIOS-Geometrie der Boot-Festplatte festgelegt ist.

message=<message-datei>

Hier wird auf eine Textdatei verwiesen, welche von LILO beim Start als erstes ausgegeben wird.

password=<passwort>

Kann sowohl am Anfang im Parameter-Abschnitt, als auch in einzelnen Systemabschnitten stehen und sichert den Zugriff auf die LILO-Dienste bzw. das Booten des betreffenden Systems mit einem Passwort ab.

read-only

Mit dieser Option weist LILO den betreffenden Kernel an, die Rootpartition zunächst nur read-only zu mounten, wie es beim Start von Linux-Systemen generell üblich ist.

delay=<zehntelsekunden>

Wenn der Prompt nicht zwingend vorgeschrieben worden ist, kann der Benutzer dennoch zur Startzeit von LILO durch Tastendruck einen Prompt anfordern. Die Delay-Option gibt die Zeit vor, die LILO während der Startzeit auf einen solchen Tastendruck wartet.

vga=<mode>

Wählt den VGA-Textmodus beim Start.

append=<parameter>

Image-Optionen für Linux-Kernel, welche die Übergabe von Kernel-Parametern ermöglichen.

Die oben beschriebenen Konfigurationsdateien stellen nur eine winzige Auswahl der tatsächlichen Möglichkeiten dar, um das System zu konfigurieren und anzupassen. So wäre es z.B. interessant gewesen, weitere Config-Files zu beschreiben, etwa die Dateien `modules.conf`, `inet.conf`, usw.... Ferner wäre eine Auseinandersetzung mit der Initialisierung der verschiedenen Run-Level ein gutes Thema gewesen, welches aber wiederum den Rahmen des Berichts gesprengt hätte.

1.4 Fazit

Der Praktikumsversuch hat gezeigt dass Linux sowohl ein sehr puristisches aber auch ein sehr anwenderfreundliches System sein kann. Die Entwickler von SuSe geben sich Mühe eine wirkliche Alternative zu Windows zu schaffen und erhalten trotzdem dem Anwender die Freiheit, selbst zu entscheiden was er möchte. Auch wenn es zu Anfang schwer fällt sich mit dieser "anderen Philosophie" vertraut zu machen, so nimmt der "Fun Faktor" mit zunehmendem Wissen zu. Auch die Dokumentation ist über die Manual Seiten sehr gut gelöst. Wenn das Grundprinzip verstanden wurde, findet man sich schnell zurecht.

Uns als Einsteiger hat das Praktikum den Ansporn gegeben, sich in dieser Richtung weiter und intensiver zu befassen. Desweiteren stellen wir uns mittlerweile die Frage, warum Linux/Unix-Systeme im Rahmen des Studiums nicht stärker in die Vorlesungen mit einbezogen werden, da sie vor allem für das Grundverständnis der Funktionen eines Betriebssystems von entscheidender Bedeutung sein können.