

Der Bootlader

Lernziele



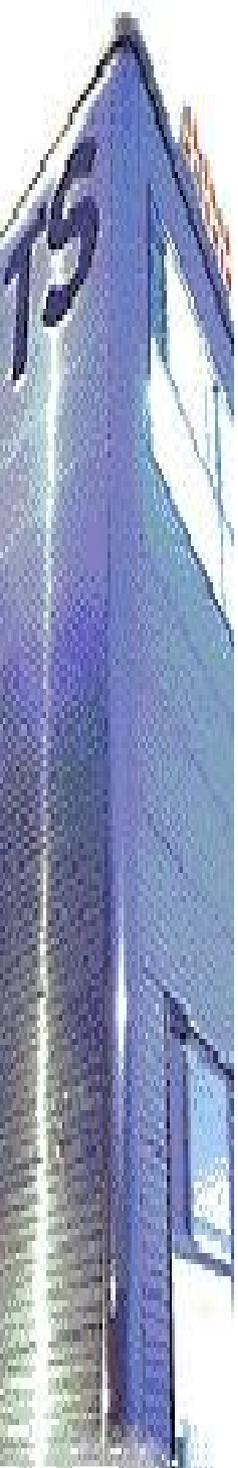
- Die Bootlader LILO und GRUB kennen und konfigurieren können
- Probleme beim Systemstart erkennen und beheben können

Was ist ein Bootlader?

- Abhängig von den Einstellungen sucht das BIOS beim Systemstart nach einem Betriebssystem
 - Diskette
 - CD-ROM
 - Festplatte
 - USB Gerät
 - Netzwerk (PXE)

Was ist ein Bootlader?

- Es werden die ersten 512 Bytes des Bootmediums ausgelesen
- Hier stehen spezielle Informationen zum System-Start
- Dieser Bereich heisst auch Master Boot Record (MBR)
- Oder generell Bootsektor
- Die ersten 446 Bytes des MBR enthalten ein Ministartprogramm



Was ist ein Bootlader?

- Im Rest des MBR folgen vier Einträge von jeweils 16 Byte
 - Für vier mögliche primäre Partitionen
 - Es handelt sich um die Partitionstabelle
- Disketten und meist auch Flash-Speicher (Memory-Sticks) enthalten keine Partitionstabelle

Was ist ein Bootlader?

- In der Regel befindet sich der Bootlader auf der Festplatte im Sektor 0 (MBR)
- Linux verwendet meist einen der Bootlader
 - LILO »Linux-Loader«
 - elilo für EFI-basierte Computer
 - GRUB »Grand Unified Bootloader«

Was ist ein Bootlader?

- Dies sind nicht nur Bootlader, sondern eigentliche Bootmanager
- Sie können über ein Menü nach Wahl
 - verschiedene Kernel booten
 - verschiedene Betriebssysteme starten



Beispiele für Bootlader

ADAM2 (Texas Instruments AR7)

Boot Camp (Apple Boot Camp)

bootman (BeOS/ZETA Boot Loader)

Bootmgr (Bootloader von Windows Vista)

Boot-US (universeller Bootmanager)

BootX

eXtended FDisk (mit Partitionierer)

Gestor de Arranque Grafico (GAG)

GRUB

LILO (Linux Loader)

Loadlin

NTLDR (NT Loader, alle Windows-Systeme von NT 3.1 bis XP/2003)

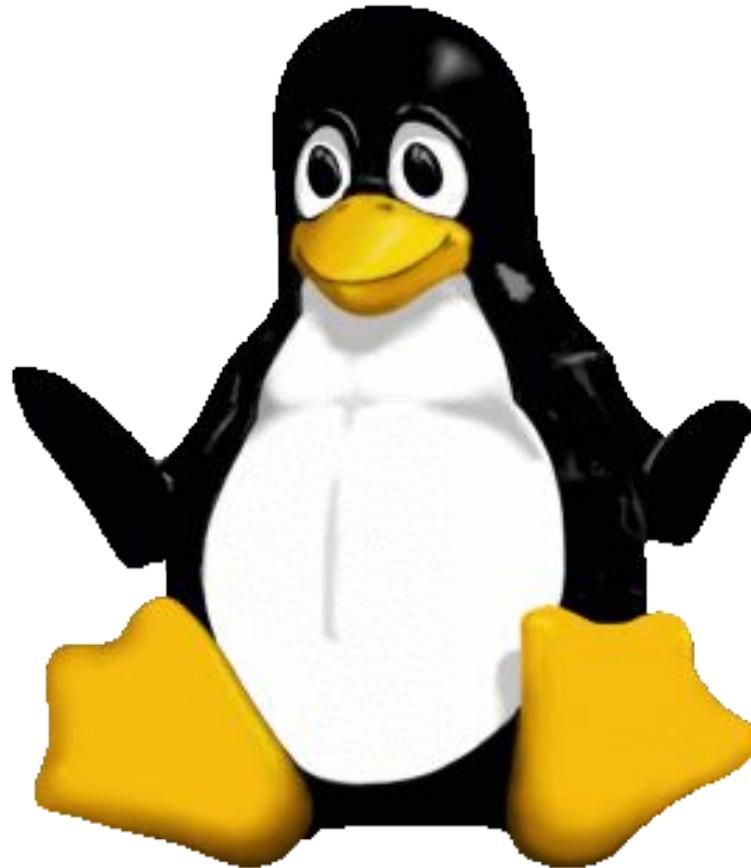
OpenBIOS

SYSLinux / **PXELinux** / **ISOLinux**

XOSL

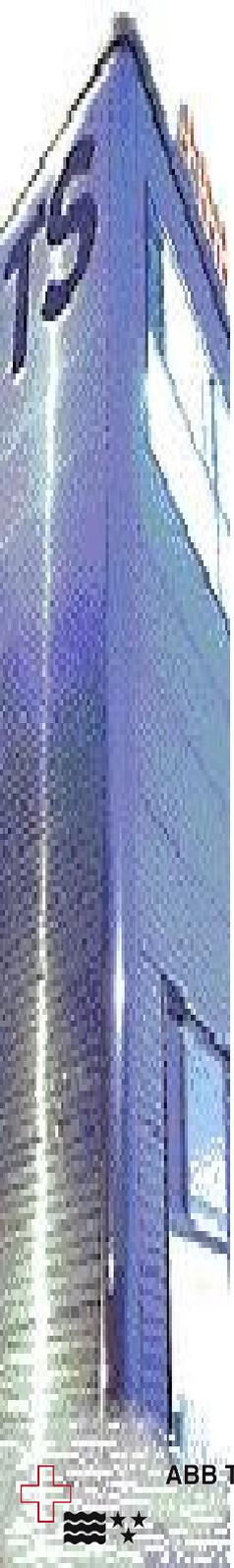
ZBOOT

Fragen?



Übungen

- [5.1] Wo auf der Festplatte darf sich der Bootlader befinden?
- Warum?



Der Linux-Loader LILO

- LILO ist der Klassiker unter den Linux-Bootladern
- Er wird allerdings von GRUB abgelöst
- LILO hat im *Real Mode* Zugriff auf die Festplatten
- Kann den Linux-Kernel laden und starten
- Kann Bootparameter übergeben
- Kann Boot-Sektoren von anderen Betriebssystemen laden und starten

Der Linux-Loader LILO

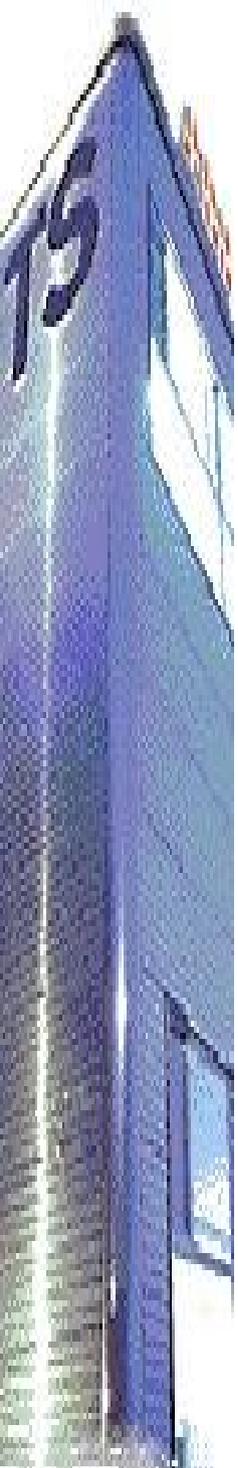
- LILO besteht aus folgenden Teilen:
 - 1 dem LILO-Bootsektor
 - erster Teil des LILO-Codes
 - 2 dem LILO-Maschinencode
 - unter `/boot/boot.b`
 - 3 einer Map-Datei
 - unter `/boot/map`
 - 4 eine Begrüßungsnachricht oder Grafik
 - unter `/boot/message`

Der Linux-Loader LILO

- ❓ Wo genau kann LILO installiert werden
 - auf einer der ersten beiden Festplatten
- in der Startphase muss der Bootlader vom BIOS erreicht werden
- Alte BIOS können nur die ersten 1024 Zylinder einer Festplatte ansprechen

Der Linux-Loader LILO

- Die Konfiguration von LILO ist in der Datei `/etc/lilo.conf`
 - i.d.R nur für den Administrator lesbar
 - enthält evtl. Kennwörter
- Nach der Modifikation von *lilo.conf*
 - muss der Bootlader
 - und evtl. die Map-Datei aktualisiert werden



Der Linux-Loader LILO

- Die Konfigurationsdatei besteht aus einem allgemeinen Teil
- und einem oder mehreren spezifischen Abschnitten
- Einen Abschnitt für jedes zu bootende System
- Das zuerst in der Liste aufgeführte Betriebssystem wird gebootet, wenn nach der vorgegebenen Wartezeit kein System ausgewählt wurde

Der Linux-Loader LILO

- Funktionen für den allgemeinen Teil der Konfigurationsdatei
 - `backup=<Datei>` ein Backup des MBR
 - `boot=<Device>` wird der MBR gespeichert
 - `delay=<Zeit>` bis default ausgeführt wird
 - `message=<Datei>` Begrüssungstext
 - `password=<Code>` unverschlüsseltes Kennwort
 - `prompt` Eingabeaufforderung
 - `timeout=<Zeit>` warten auf Eingabe

Der Linux-Loader LILO

● Funktionen für den allgemeinen Teil der Konfigurationsdatei

● vga=<Modus> wählt den VGA-Textmodus beim Starten

● normal für 80 × 25 Zeichen

● ext für 80 × 50 Zeichen

● ask für Nachfrage beim Booten

● ... andere

Der Linux-Loader LILO

- Funktionen für den spezifischen Teil der Konfigurationsdatei
 - `image=<Kernel>` die Kernel Datei incl. Pfad (z. B. `/boot/vmlinuz`)
 - `label=<Name>` Bezeichnung (15 Zeichen)
 - `root=<Partition>` die Systempartition z.B. `/dev/hda2`
 - `initrd=<Datei>` initiale Ramdisk
 - `other=<Gerät>` Nicht-Linux Betriebssystem
 - `loader=<Datei>` meist `/boot/chain.b`

LILO ändern

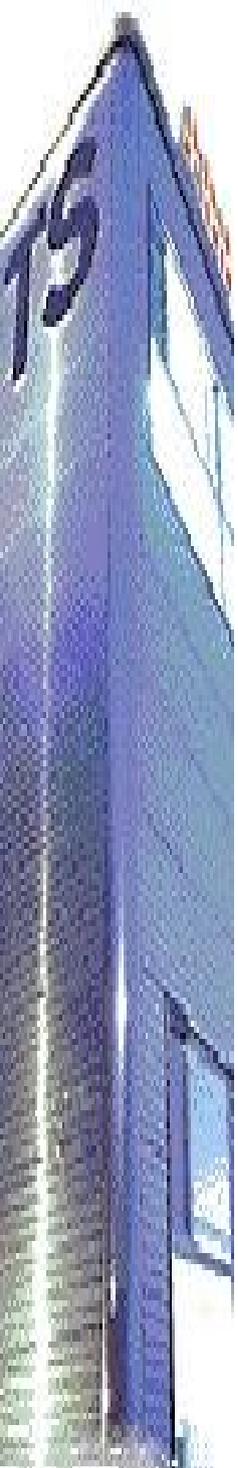
- Nach jeder Änderung der LILO-Konfiguration muss LILO neu installiert werden
- Nach Ändern der *lilo.conf*
- Nach Änderungen in der *initrd*
- Nach Änderungen am Menu
- # /sbin/lilo
- Der MBR wird neu geschrieben

Der »Grand Unified Bootloader« GRUB

- Bei vielen Distributionen ist GRUB der Standard-Bootlader
- Moderne Zusatzfunktionen, im Vergleich zu LILO
- GRUB besteht wie LILO eigentlich aus zwei Programmen:
 - 1 Die 446 Byte grosse Bootroutine im MBR
 - 2 eine umfangreichere Routine, die von der ersten Stufe geladen wird

Der »Grand Unified Bootloader« GRUB

- GRUB unterstützt schon vor dem Systemstart einen Zugriff auf Dateisysteme
- Keine Veränderung des MBR nötig, wenn die GRUB-Konfiguration oder der Kernel verändert werden
- GRUB bietet schon vor dem Start eine GRUB-Shell mit diversen Befehlen an
- Es können auch beliebige nicht in der Konfigurationsdatei stehende Systeme geladen werden



Die Konfiguration von GRUB

- Die zentrale Konfigurationsdatei befindetet sich in der Regel unter
 - # /boot/grub/menu.lst
- Hier werden die Grundeinstellungen vorgenommen
- Die einzelnen zu bootenden Betriebssysteme festgelegt
- Die Kernel-Parameter konfiguriert

Die Konfiguration von GRUB

Die Datei könnte aussehen wie folgt:

```
default 1
timeout 10

title linux
    kernel (hd0,1)/boot/vmlinuz root=/dev/hda2
    initrd (hd0,1)/boot/initrd

title failsafe
    kernel (hd0,1)/boot/vmlinuz.bak root=/dev/hda2 apm=off acpi=off
    initrd (hd0,1)/initrd.bak

title einanderessystem
    root (hd0,2)
    makeactive
    chainloader +1

title floppy
    root (fd0)
    chainloader +1
```

Die Konfiguration von GRUB

● Die einzelnen Parameter bedeuten:

● default

Gibt das standardmässig zu bootende System an

● timeout

So lange wird das GRUB-Menü angezeigt

● title

Betriebssystemeintrag und Namen im GRUB-Menü

● kernel

Gibt den zu bootenden Linux-Kernel und die benötigten Parameter an

Die Konfiguration von GRUB

- ! Die 0. Festplatte ist die 1. Festplatte in der BIOS Bootreihenfolge
- (hd0,1)/boot/vmlinuz bedeutet, dass der Kernel in /boot/vmlinuz auf der 2. Partition der 1. Festplatte zu finden ist (/dev/hda2)
- Es gibt keine Unterscheidung zwischen IDE und SCSI
- Die genaue Zuordnung der einzelnen Laufwerke steht in der Datei
 - # /boot/grub/device.map

Die Konfiguration von GRUB

-  `initrd` Gibt den Ort der Initial-RAM-Disk an, wo die Kernelmodule stehen
-  `root` Legt für Fremd-Systeme die Systempartition fest
-  `chainloader +1` Bezeichnet einen fremden zu ladenden Bootlader
-  `makeactive` Macht die angesprochene Partition temporär bootfähig
-  `hide, unhide` Verstecken einer Partition
-  `map` Vertauschen der Festplatten

Die Installation von GRUB

- Eine Neuinstallation von Grub ist nur selten nötig
 - Bei Änderungen an der Datei *device.map*
 - Wenn der Kernel umbenannt wurde
 - Wenn der MBR überschrieben wurde
-
- Änderungen an der Konfiguration werden direkt beim nächsten Systemstart übernommen

Die Installation von GRUB

- Für die Installation des MBR wird eine GRUB-Shell verwendet

- Aufruf von Grub

 - # grub

- Einfacher gehts im Batchmode

 - # grub --batch -device-map=/boot/grub/device.map\
< /etc/grub.conf

- Auch das Install-Scribt erledigt das

 - # grub-install /dev/hda

Die Installation von GRUB

- Der Installer verwendet die Konfig aus `/etc/grub.conf`

```
root (hd0,1)
install /grub/stage1 d (hd0)(hd0,1) /boot/grub/menu.lst
quit
```

- root* ist die Partition, auf der sich das Grubverzeichnis befindet
- install* installiert die erste Stufe in den MBR vom 1. Harddisk (hd0)
- Das Menu liegt auf Partition 2 in `/boot/grub/`

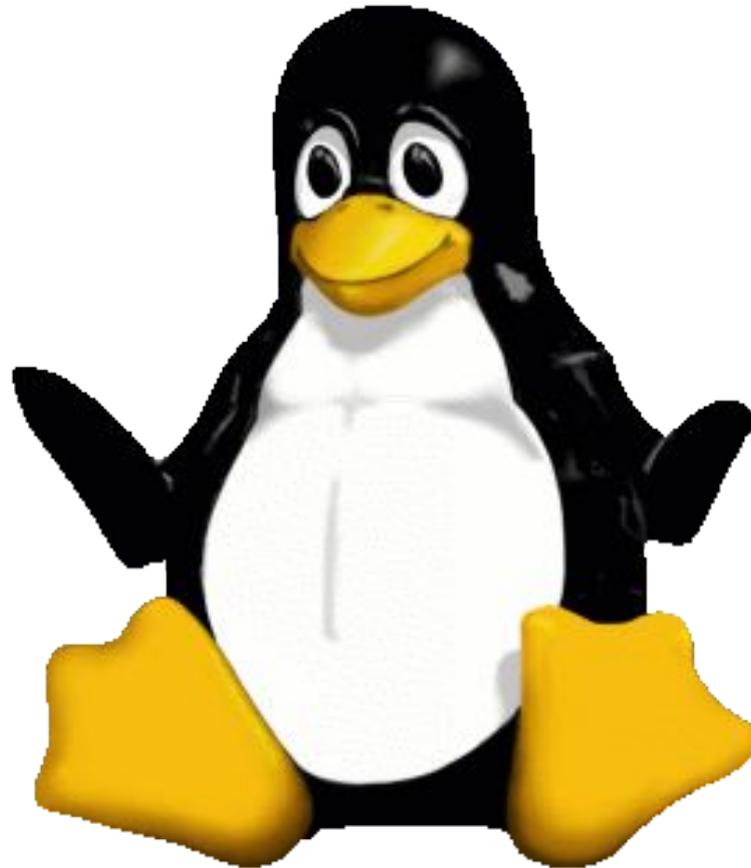
Sicherheitsaspekte von GRUB

- Die GRUB-Shell bietet viele Möglichkeiten
 - Ohne root-Kennwort auf die Dateisysteme zugreifen!
 - Eingabe von Bootparametern
 - Man kann auch in eine Shell booten
- Kennwort Abfrage durch setzen eines zentralen Kennwortes in *menu.lst*
- Kennwort für einen bestimmten Menüpunkt in der Auswahl

Sicherheitsaspekte von GRUB

- Das Kennwort wird im zentralen Teil von *menu.lst* eingetragen
- Ein verschlüsseltes Kennwort erzeugt
 - # grub-md5-crypt
- In der GRUB-Shell
 - # md5crypt
- Für die Kennwortabfrage eines Menüpunkts wird der Eintrag *lock* dem entsprechenden Abschnitt hinzugefügt

Fragen?



Übungen

- ❓ [5.2] Welches ist die Konfigurationsdatei für Ihren Bootloader?
- 🕒 Erstellen Sie einen neuen Eintrag, mit dem Sie ein weiteres Betriebssystem starten könnten.
- 🕒 Machen Sie vorher eine Sicherungskopie der Datei.

Übungen

- ❓ [5.3] Verhindern Sie, dass ein normaler Benutzer unter Umgehung von init direkt in eine Shell booten kann.
- ❓ Wie generieren Sie eine Kennwortabfrage beim Booten für ein bestimmtes Betriebssystem?

Entfernen des Bootladers

- Die Deinstallation eines Bootladers im MBR ist ein heikler Eingriff ins System
- Vorher ist sicherzustellen, dass das System im Zweifelsfall auch von Diskette oder CD gestartet werden kann
- Bei LILO löscht man den MBR mit
 - `lilo -u`
 - Der gesicherte Original-Bootlader wird zurück geschrieben

Entfernen des Bootladers

- Grub unterstützt kein Deinstallieren des MBR
- Um den Bootlader vom MBR der ersten IDE-Platte zu löschen, kann man *dd* verwenden
 - `dd if=/dev/zero of=/dev/hda bs=446 count=1`
 - *dd* schreibt binäre Nullen an die Stelle des Bootloaders
 - `xxd -g1 -l 512 /dev/hda`

Übungen

 [5.4] Wie lässt sich ein Backup des Bootladers erstellen?

Kernelparameter

- Linux kann vom Bootlader eine Kommandozeile übernehmen
- Hierfür dient bei LILO die Option *append=...*
- Bei GRUB können die Parameter einfach dem Kernel-Eintrag folgen
- Alternativ können die Parameter auch am Boot-Prompt eingegeben werden

Kernelparameter

- Es gibt verschiedene Arten von Argumenten
- ① Die eingestellten Parameter werden ersetzt
 - z.B. *root*, *rw*
- ② Konfiguration bestimmter Gerätetreiber
- ③ Argumente für allgemeine Einstellungen
 - z.B. *init* oder *reserve*

Kernelparameter

Einige typische Argumente:

- ro Die Root-Partition wird read-only gemountet
- init=<Programm> Startet z.B. /bin/bash
- <Runlevel> Startet in den angegebenen Run-Level
- acpi=off Startet ohne ACPI-Bios Zugriff
- vga=normal Standard VGA Ausgabe
- selinux=0 SE-Linux wird nicht gestartet

Kernelparameter

- Codetabelle für VGA Framebuffer:

- <http://linuxwiki.de/GRUB>

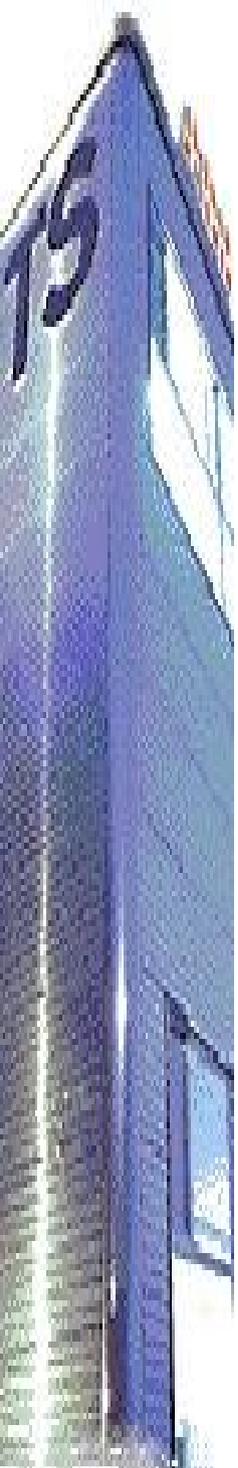
- gängige Werte sind:

- 640x480 0x311 (785)

- 800x600 0x314 (788)

- 1024x768 0x317 (791)

- Nicht alle bootparameter funktionieren überall



Probleme beim Systemstart

- Der Bootlader ist defekt oder falsch konfiguriert
- Der Kernel ist defekt oder wurde falsch konfiguriert
- Hardwareprobleme
- Die Partitionstabelle im MBR ist fehlerhaft
- Fehler in Startup-Skripts
- Festplatte oder Dateisystem sind defekt