

DER BOOTLOADER GRUB UND SEINE KONFIGURATION

STARTHILFE

Der wohl flexibelste Bootloader entspringt dem Linux-Umfeld: GRUB. Dieser bekannte Bootmanager ist aber nicht auf das Pinguin-System beschränkt, sondern leistet auch gute Dienste bei der Verwaltung mehrerer Windows-Installationen.

VON **MATTIAS SCHLENKER**

Billig sollte sie sein, die Firmware eines gewöhnlichen PC. Das Resultat kennen wir alle: Im Gegensatz zur Firmware klassischer Workstations, die Dateisysteme lesen können und die Wartung über das Netzwerk erlauben, blieb das BIOS „dumm“ – die ersten 446 Bytes von Diskette oder Festplatte mussten und müssen genügen, um die Information unterzubringen, wo der PC sein Betriebssystem findet. Ein Vorteil dieser Einfachheit ist die hohe Flexibilität, die es auch erlaubt, verschiedenste Betriebssysteme auf einem Rechner unterzubringen, die jeweils ihren eigenen Bootloader mitbringen. Möchte man mehrere Systeme auf einem Rechner betreiben, muss

jedoch ein Bootmanager her, der nicht nur die Eigenheiten der zu startenden Systeme kennt, sondern auch mit vielen größeren und kleineren Besonderheiten verschiedener BIOSe klarkommt.

Solch ein Bootloader ist der im Umfeld des GNU-Projektes entstandene „Grand Unified Bootloader“, kurz GRUB. GRUBs Funktionsumfang ist eindrucksvoll: Linux-Kernel und die Kernel einiger Unix-Systeme bootet er direkt und findet sie anhand ihrer Lage im Dateisystem, anstatt wie LILO über die Block-Adresse auf sie zuzugreifen. Das erlaubt im Notfall den Start von Kernen, die dem Bootloader noch nicht bekannt gemacht wurden.

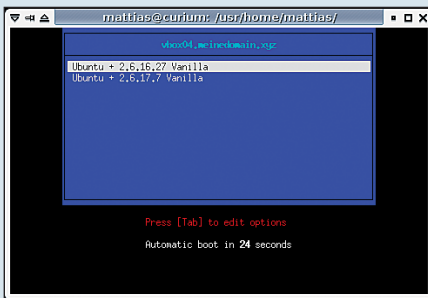
Daneben kann GRUB beim Start kleinere Modifikationen an der Partitionstabelle vornehmen. Insbesondere, wenn verschiedene Windows-Installationen parallel vorgehalten werden, kann die gerade gestartete Partition „aktiviert“ und eine andere versteckt werden. Selbstverständlich arbeitet GRUB als Chainloader, das heißt, er startet die Bootloader anderer Systeme, vor allem Windows und FreeBSD. Sogar die Tücken manch eines modernen BIOS kann GRUB umschiffen, indem er die Reihenfolge von Festplatten vertauscht, was gelegentlich notwendig ist, wenn das BIOS IDE-, SCSI- und SATA-Platten anders anordnet, als es der Benutzer wünscht.

Simpel, aber effizient: SYSLINUX

Der komplexe Aufbau von GRUB hat nicht nur Vorteile. Größter Nachteil ist, dass die Reihenfolge der Festplatten bereits bei der Installation des Bootloaders festgelegt sein muss. Beim Start von USB-Stick kann es jedoch vorkommen, dass ein BIOS den Stick als Floppy (*fd0*) erkennt, ein anderes ihn als (*hd0*) – auf einem der Systeme würde der Start versagen.

Soll primär Linux gebootet werden, bietet sich deshalb ein Blick auf die Bootloader der *SYSLINUX*-Familie an. Alle diese Loader unterscheiden sich von GRUB dadurch, dass sie in die Partition anstatt in den MBR installiert werden. Eine Einschränkung besteht darin, dass sie Kernel und Initrd auf der Partition erwarten, auf welcher der Loader installiert ist.

Die SYSLINUX-Familie besteht aus:



EXTLINUX ist eine schlanke Alternative zu GRUB, die vor allem auf Servern Sinn macht.

- SYSLINUX für FAT16-Partitionen, primär für USB-Sticks und -Festplatten
- ISOLINUX für Linux-Live- und Installations-CDS
- EXTLINUX für EXT2/3-Partitionen
- PXELINUX für den Start festplattenloser Linux-Systeme

SYSLINUX ist bei USB-Sticks praktisch Standard, die Verwendung erklärt die Webseite www.pc-magazin.de/praxis/linux/a/LINUX_VON_USB_STICK_UND_FESTPLATTE_BOOTEN. Auch ISOLINUX wird bei vielen Distributionen als Standard-Bootloader verwendet, teils gut versteckt hinter hübschen Splash-Screens. EXTLINUX gilt als Geheimtipp, er kann jedoch gerade bei Servern, auf denen sowieso eine separate Boot-Partition verwendet wird und oft eine serielle Konsole erwünscht ist, mit seiner einfachen Konfiguration gute Dienste leisten. Ein Installationsbeispiel, das Boot-Menüs und die serielle Konsole berücksichtigt, finden Sie auf der Webseite des Autors blog.rootserverexperiment.de/2006/08/01/extlinux-flexibler-bootloader-fur-den-rootserver

Zweigeteilt

Eine Besonderheit von GRUB ist, dass er seinen gesamten Funktionsumfang nur dann voll zur Geltung bringen kann, wenn er in den Master Boot Record der Festplatte installiert wird. Windows und DOS schreiben in den MBR lediglich einen Standard-Bootrecord, der das BIOS anweist, eine in der Partitionstabelle aktiv markierte Partition zu suchen und deren Bootloader zu starten. Da GRUB nicht seinen gesamten Code auf 446 Bytes unterbringen kann, schreibt er seine zweite Stufe in die Installationspartition. Sie sollten dies im Hinterkopf behalten, wenn Sie öfters Partitionen verschieben, löschen und neu anlegen – verschiebt man die Partition, auf der GRUB installiert wurde, bootet der PC nicht mehr von Festplatte.

Der einfachste Weg, die Konfiguration von GRUB kennen zu lernen, ist Ubuntu Linux oder SUSE auf einen gewöhnlichen PC mit einer einzigen IDE- oder SATA-Festplatte zu installieren und während der Installation das Überschreiben des Master-Boot-Records zu erlauben. Der Installer erstellt dann eine Konfigurationsdatei */boot/grub/menu.lst* und eine weitere Datei */boot/grub/device.map*, in der die Zuordnung der Festplatten steht. Auf dem Testrechner mit drei primären Partitionen (1. NTFS, 2. Linux Swap, 3. Linux Root) erhielten wir dabei folgende */boot/grub/menu.lst*-Kommentare. Die deutsche Tastaturbelegung im Boot-Editor haben wir entfernt (s. Listing 1):

Die dazu gehörige */boot/grub/device.map* enthielt nur einen Eintrag:

(hd0) /dev/sda

Grub beginnt die Zählung von Partitionen und Festplatten bei Null, (*hd0*) bezeichnet die erste vom BIOS erkannte Festplatte – in der Regel die Platte, die in der Boot-Reihenfolge an erster Stelle steht. (*hd0,0*) steht demnach für die erste Partition der ersten Festplatte, hier die NTFS-Partition mit einer Vista-Installation.

Der erste Eintrag steht für die Linux-Installation. Kernel und Initrd werden direkt über ihren Pfad auf der 3. Partition auf der ersten Festplatte aufgerufen. Das Schlüsselwort *boot* schließt den Eintrag ab. Es folgt Windows auf der ersten Partition der ersten Festplatte. Da Vista einen eigenen Loader am Anfang seiner Partition mitbringt, wird dieser über das Schlüsselwort *chainloader* aufgerufen, *makeactive* markiert die Boot-Partition als *aktiv*.

Listing 1

```
default 0
timeout 10

title          Ubuntu, Kernel 2.6.17
kernel         (hd0,2)/boot/vmlinuz root=/dev/sda3 ro
initrd         (hd0,2)/boot/initrd.img
boot

title          Windows Vista
root           (hd0,0)
makeactive
chainloader    +1
boot
```

GRUB installieren

Aktuelle Linux-Distributionen erledigen die Installation von GRUB automatisch. Hat eine erneute Windows-Installation den Master-Bootrecord überschrieben, werden Sie GRUB jedoch neu installieren müssen – ebenso, wenn Sie GRUB dazu nutzen, verschiedene Windows-Installationen komfortabel zu booten, ohne dass eine Linux-Installation auf dem Rechner vorhanden ist. Dies geht am komfortabelsten mit einer Linux-Live-CD wie Knoppix oder Kanotix. Starten Sie von dieser und mounten Sie die Partition, auf der Sie GRUB installieren wollen. Öffnen Sie an-

schließlich eine Root-Shell und führen Sie den GRUB-Installer aus:

```
grub-install --root-directory=  
→ /media/hda3 --recheck /dev/hda
```

Dieser Befehl installiert GRUB in den Master-Bootrecord der ersten IDE-Festplatte und die zweite Stufe auf die dritte Partition dieser Platte.

Der Parameter `--recheck` prüft die Zuordnung zwischen Linux-Laufwerksbezeichnungen und gibt sie am Ende aus. Wird die Boot-Platte nicht korrekt als (`hd0`) identifiziert, was häufig der Fall ist, wenn SATA- und IDE-Geräte in einem Rechner eingebaut sind, müssen Sie die Datei `/boot/grub/device.map` anpassen und den Befehl erneut ohne `--recheck` ausführen.

Enthielt die Partition, auf der GRUB installiert wurde, noch keine `/boot/grub/menu.lst`, muss diese erstellt werden, um beim Systemstart nicht in der Eingabeaufforderung `grub>`

zu landen und Bootpartition, Kernel und Initrd oder Chainloader von Hand eintippen zu müssen.

Profi-Bootloader

GRUB bringt einige Features mit, die den Bootloader für Firmenumgebungen und Server attraktiv machen. Eines davon ist die Möglichkeit, bestimmte Boot-Einträge durch ein Passwort abzusichern. So kann das Abändern von Menüeinträgen genauso verhindert werden wie die Anwahl gesperrter Optionen. Dies erlaubt es, Rettungssysteme so abzusichern, dass sie nur der Administrator anwählen kann. Erstellen Sie den Passwort-Hash unter Linux (oder unter Windows in einer Cygwin-Umgebung) mit dem Befehl

```
openssl passwd -1
```

Listing 2

```
default 0
timeout 10
password --md5 $1$0G9J9Le6$wfoIs2sHJR8rkzS4VjT5NO

title          Ubuntu, Kernel 2.6.17
kernel        (hd0,2)/boot/vmlinuz root=/dev/sda3 ro
initrd        (hd0,2)/boot/initrd.img
lock
boot

title          Windows Vista
root          (hd0,0)
makeactive
chainloader  +1
boot
```

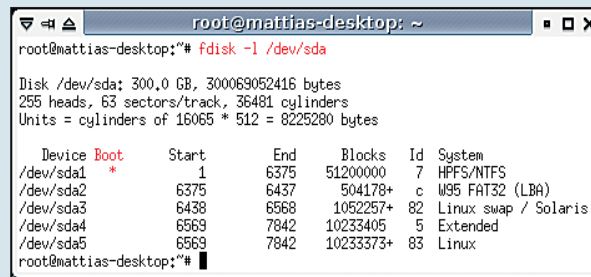
Der ausgegebene Passwort-Hash wird vor die Menüeinträge in die Konfigurationsdatei eingegeben. Der Befehl `lock` sperrt einen Menüeintrag schließlich ganz, andere Menüeinträge können ausgewählt und gestartet, aber nicht ohne Passwort verändert werden (s. Listing 2). Eine weitere interessante Erweiterung ist

die Möglichkeit, GRUB mit Treibern für Netzwerkkarten und einem DHCP- sowie TFTP-Client zu kompilieren. Dies erleichtert das Zurückspielen von Betriebssystem-Images via Netzwerk, auch wenn ein Rechner nicht für den Boot-Vorgang per PXE konfiguriert wurde.

Tausche und verstecke

Mehrere Windows-Installationen auf einem Rechner vorzuhalten, kann für einiges Kopfzerbrechen sorgen, da das System aus Redmond bei der Neuinstallation gerne den Bootloader einer bereits vorhandenen Windows-Installation mitbenutzt. Löscht man später die erste vorhandene Windows-Partition, bleiben weitere oft *unbootable*. Hier kann GRUB durch das geschickte Verstecken

GRUB loswerden



```
root@mattias-desktop:~# fdisk -l /dev/sda

Disk /dev/sda: 300,0 GB, 300069052416 bytes
255 heads, 63 sectors/track, 36481 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device  Boot      Start         End      Blocks   Id  System
/dev/sda1  *          1         6375     5120000    7  HPFS/NTFS
/dev/sda2             6375         6437      504178+    c  W95 FAT32 (LBA)
/dev/sda3             6438         6568     1052257+   82  Linux swap / Solaris
/dev/sda4             6569         7842     10233405    5  Extended
/dev/sda5             6569         7842     10233373+   83  Linux
root@mattias-desktop:~#
```

➤ Soll die Linux-Partition, auf der GRUB und dessen Konfiguration installiert ist, verschoben oder gelöscht werden, ist Vorsicht angebracht: Bleibt GRUBs erste Stufe im MBR und es fehlen die weiteren Stufen, ist das gesamte System nicht mehr bootbar. Sie sollten deshalb zuerst GRUB entfernen und dann die nicht mehr benötigten Partitionen löschen. Auch für die Entfernung von GRUB ist eine Linux-Live-CD wie Knoppix das geeignetste Werkzeug:

1 Booten Sie von der Live-CD und identifizieren Sie mit `sudo fdisk -l`

Ihre Systemplatte. In der Regel wird `/dev/sda` oder `/dev/hda` das erste Boot-Laufwerk sein.

2 Schreiben Sie mit `sudo ms-sys -m /dev/hda`

einen neuen Windows-XP-Boot-Sektor in den Master-Bootrecord. Sollten Sie kein Windows XP starten wollen, zeigt `ms-sys -h` die Boot-Sektoren, die das Tool installieren kann.

3 Achten Sie darauf, dass keine Partition der zu modifizierenden Platte gemountet ist (`df`). Starten Sie dann mit `sudo gparted`

das grafische Partitionierungswerkzeug und setzen Sie auf der Windows-XP-Partition das Flag *bootfähig* (manchmal als *bootable* oder *active* bezeichnet).

4 Sie können bereits in `gparted` die Partition löschen, auf der GRUB installiert war und den freigewordenen Platz einer neuen Partition zuweisen oder die Partition davor entsprechend vergrößern.

Handelt es sich um eine Standard-Linux-Installation, die Sie von Festplatte entfernen wollen, genügen auch die Kommandos `fixmbr` und ggf. `fixboot` in der Rettungskonsole der Windows-Installations-CD/-DVD. Nach dem Herunterfahren bootet Windows wieder ohne vorgelagerten GRUB, und Änderungen an der Partitionierung haben keine unmittelbaren Folgen auf den Systemstart mehr.

Fallbeispiel: XP und Vista auf verschiedenen Festplatten

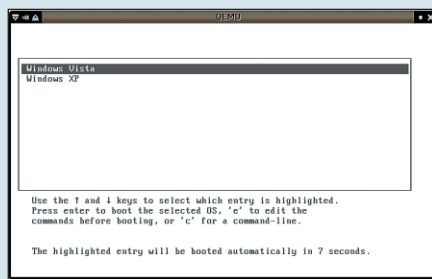
➤ Auf einem Testrechner sollten Windows XP und Windows-Vista auf verschiedenen Festplatten nebeneinander residieren, ohne sich gegenseitig zu beeinflussen. Windows XP befand sich bereits auf einer 80 GByte großen IDE-Festplatte, die wir nicht weiter verkleinern wollten. Wir entschlossen uns daher für den Einbau einer SATA-Festplatte für Windows Vista und den Bootloader. Dafür gingen wir wie folgt vor:

1 Wir entfernten zunächst die IDE-Festplatte, auf der XP installiert war, und bauten die SATA-Festplatte ein. So konnten wir sicherstellen, dass jedes Windows nichts vom anderen weiß.

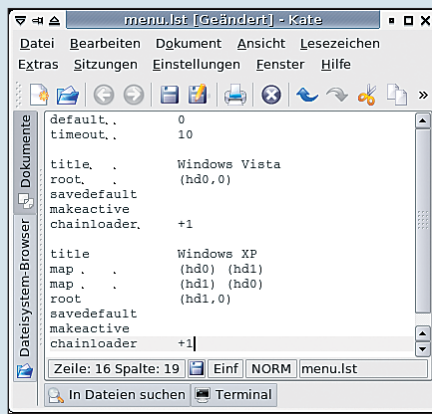
2 Bei der Vista-Installation war nichts weiter zu beachten, außer dass etwas Platz frei blieb, um den Bootloader darauf installieren zu können. Wir ließen genug Platz frei, um später eine komplette Linux-Installation vornehmen zu können.

3 Anschließend starteten wir Knoppix und legten mit `qparted` hinter der Vista-Partition eine kleine primäre Linux-Partition (`ext3`) an. Alternativ können Sie eine FAT32-Partition verwenden.

4 Die Linux-Partition mounteten wir und installierten GRUB:



GRUB leistet gute Dienste um Windows Vista gemeinsam mit XP auf einem Rechner unterzubringen.



Damit das auf der (zuerst eingebauten) IDE-Festplatte installierte XP startet, werden mittels `map` die BIOS-Zuordnungen der Laufwerke vertauscht.

```
mount /dev/sda2 /media/sda2
grub-install --root-directory=
↳ /media/sda2 --recheck /dev/sda
```

5 In der GRUB-Konfigurationsdatei musste beim Eintrag für XP – dessen Festplatte nun ja die zweite war – der Festplattentausch aktiviert werden:

```
default 0
timeout 10

title Windows Vista
root (hd0,0)
savedefault
makeactive
chainloader +1

title Windows XP
map (hd0) (hd1)
map (hd1) (hd0)
root (hd1,0)
savedefault
makeactive
chainloader +1
```

6 Nach dem Herunterfahren des Knoppix-Livesystems steckten wir die IDE-Festplatte wieder an und starteten den Rechner: Beide Boot-Einträge funktionierten problemlos und kein Windows beeinflusste das andere.

und Sichtbarmachen von Partitionen sowie das Vertauschen von Festplatten abhelfen. Die notwendigen Befehle werden als Teil des jeweiligen Menüeintrages untergebracht:

```
hide (hd0,0)
unhide (hd0,1)
Dieser Eintrag versteckt die erste Partition der ersten Festplatte und macht die zweite sichtbar, während
map (hd0) (hd1)
map (hd1) (hd0)
```

die erste und die zweite Festplatte vertauscht.

Rückkehr der Diskette

Moderne NT-basierte Systeme aus Redmond und die freien Unix-Systeme haben das Boot-Medium Diskette ziemlich alt aussehen lassen: 1,44 Megabyte reichen nur mit argen Klimmzügen für Kernel und Ramdisk. GRUB kann der Floppy jedoch zu einer neuen Daseinsberechtigung verhelfen. Der Clou: Da GRUB Dateisysteme lesen kann und deshalb nicht darauf angewiesen ist, den Kernel auf der Bootloader-Partition vorzufinden, ist es möglich, eine Diskette zu erstellen, die nur

den Bootloader, aber keinen Systemkern enthält. Mit dieser Floppy können anschließend bei defektem MBR der Festplatte der Kernel und die ursprüngliche GRUB-Konfiguration gesucht und das System gestartet werden. Die Bootfloppy lässt sich ebenfalls am leichtesten unter Linux, beispielsweise Knoppix, erstellen. Mounten Sie eine DOS-formatierte Diskette und installieren Sie GRUB auf diese. Der als *root-Directory* angegebene Mountpoint mag sich von *Distri* zu *Distri* unterscheiden. Im Zweifel zeigt ihn `dfan`:

```
grub-install --recheck --root-directory=
↳ /media/fd0 /dev/fd0
```

Wenn Sie von dieser Diskette booten, erhalten Sie die GRUB-Shell. In dieser können Sie sich Verzeichnisinhalte und Partitionierungsdaten auflisten lassen. Eine Übersicht der Befehle liefert `help`. Um eine Linux-Installation zu booten, genügt die Angabe von Kernel und `initrd`, gefolgt vom Befehl `boot`:

```
grub> kernel (hd0,2)/boot/vmlinuz
↳ root=/dev/sda3 ro
Grub> initrd (hd0,2)/boot/initrd.img
Grub> boot
```

Wahrscheinlich haben Sie nicht alle Boot-Parameter und die exakte Lage Ihrer Kernel im Kopf. Es ist deshalb ratsam, auf der Diskette eine `boot/grub/menu.lst` anzulegen. Das kann eine Kopie der `menu.lst` des Systems sein, für das die Floppy vorgesehen ist, oder Sie kombinieren die Menüeinträge verschiedener Rechner zu einer großen Konfigurationsdatei. In diesem Fall sollten Sie im Namen des jeweiligen Eintrages einen Hinweis auf die betreffende Maschine unterbringen.

Fazit

Paradoxerweise ist der Linux-Kernel mittlerweile so einfach zu handhaben, dass GRUB seine Stärken bei einem Rechner ohne weitere Betriebssysteme kaum ausspielen kann. Erst wenn die Beschränkungen der PC-Architektur greifen, kommt GRUBs Vielseitigkeit voll zur Geltung. Ob noch ein Linux an Board ist, ist dabei zweitrangig: Unser Beispiel mit Vista und XP auf verschiedenen Festplatten zeigt, dass GRUB auch typische Windows-Probleme auf eine einfache Art und Weise lösen kann.

jkn