

2. Init-Prozess/ Bootloader / Bootconfig-Dateien

- Phasen des Bootprozesses
 - Hardwareboot / MBR
 - Bootcode / Partitionstabelle
 - Bootloader
 - Kernelstart
 - Init und Runlevel

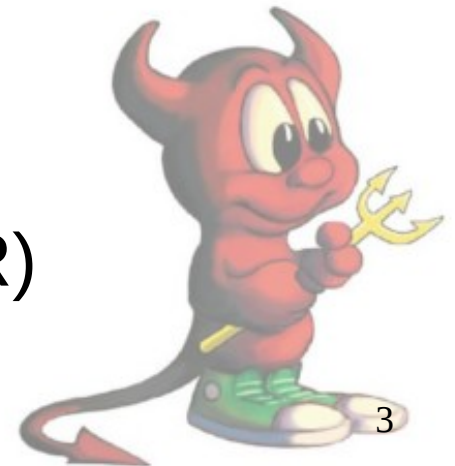


Der Bootstrap des Münchhausen



Der Hardwareboot

- Basic Input-Output System → B.I.O.S.?
- PowerOn Selftest → P.O.S.T. ?
- Suche nach bootbaren Medium
- Hardwarespezifische Einstellungen im B.I.O.S.
- Aufbau des M.B.R. (hier erster Sektor der FP):
 - Bootloader
 - Partitionstabelle (64 Byte)
 - Magic Number (steht am Ende des MBR)



Der Master Boot Record

E MBR

```
0000 fa e8 42 00 7b 01 4c 49 4c 4f 01 00 16 01 00 00
0010 00 00 00 00 57 b9 e5 01 91 00 00 00 ff ff c0 69
0020 e0 c0 02 bc 69 e0 c0 02 bd 69 e0 c0 02 be 69 e0
0030 c0 02 bb 69 e0 c0 02 01 c2 69 e0 c0 02 cf e1 e0
0040 cb 02 00 00 00 00 58 8c cf c1 e8 04 01 f8 8e d0
0050 bc 00 08 fb 52 53 06 56 fc 8e d8 8e c0 b0 0d e8
0060 83 00 b0 0a e8 7e 00 b0 4c e8 79 00 be 3d 00 bb
0070 00 02 53 e8 1a 00 5e cd 12 c1 e0 06 2d 20 03 50
0080 07 31 db e8 0a 00 75 fb b0 49 e8 58 00 06 53 cb
0090 ad 91 ac a8 60 75 0f 4e ad 89 c2 09 c8 74 26 ac
00a0 b4 02 cd 13 eb 1a 92 ad f6 c2 20 75 02 30 e4 97
00b0 f6 c2 10 74 08 03 0e 10 00 13 3e 12 00 e8 2e 00
00c0 72 04 80 c7 02 c3 b0 20 e8 1a 00 e8 07 00 31 c0
00d0 cd 13 58 eb 97 c1 c0 04 e8 03 00 c1 c0 04 24 0f
00e0 27 04 f0 14 40 50 30 ff b4 0e cd 10 58 c3 56 51
00f0 53 88 d3 80 e2 8f f6 c3 20 74 2e bb aa 55 b8 00
0100 41 cd 13 72 24 81 fb 55 aa 75 1e f6 c1 01 74 19
0110 5b 59 66 6a 00 57 51 06 53 6a 01 6a 10 89 e6 b8
0120 00 42 cd 13 8d 64 10 eb 45 5b 59 53 52 57 51 06
0130 b4 08 cd 13 07 72 39 51 c0 e9 06 86 e9 89 cf 59
0140 c1 ea 08 92 40 83 e1 3f f7 e1 96 58 5a 39 f2 73
0150 23 f7 f6 39 f8 77 1d c0 e4 06 86 e0 92 f6 f1 fe
0160 c4 00 e2 89 d1 5a 5b 86 f0 b8 01 02 cd 13 eb 09
0170 59 5f eb 02 b4 40 5a 5b f9 5e c3 00 00 00 00 00
0180 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01b0 00 00 00 00 00 00 48 6e 0f 81 b1 37 00 00 00 01
01c0 01 00 83 fe ff ff 3f 00 00 00 34 0a 80 02 80 fe
01d0 ff ff 83 fe ff ff 73 0a 80 02 73 0a 80 02 00 fe
01e0 ff ff 83 fe ff ff e6 14 00 05 97 24 40 04 00 fe
01f0 ff ff 82 fe ff ff 7d 39 40 09 44 ab 10 00 55 aa
```



Bootcode (Bootloader)

- Besteht aus Code zum Laden des B.S. oder weiteren Bootcodes (z.B. GRUB-Verzeichnis)
- Damit von einem Medium gebootet werden kann, muss dieser Code gültig sein
- Der Initiale Bootloader wird (meist) aus einer Referenz zum M.B.R. geladen
- Der Bootloader ist in den ersten 446 Byte untergebracht (z.B. GRUB referenziert, weil er größer ist)
- Die nächsten 64 Byte sind für Partitionstabelleneinträge belegt



Die Partitionstabelle

- Besteht aus vier 16 Byte langen Einträgen (stellv. für die max. 4 primären Festplattenpartitionen)
- Ein Partitionseintrag enthält:
 - ID
 - Partitionssektor (Anzahl, Beginn- und Endsektor)
 - 83 - ext2/3
 - 0b – F.A.T.32
 - 07 – N.T.F.S.



Bootmanager

- Beispiel GRUB:



Bootmanager

- Realisierung des Betriebssystemstart
- Er startet den Kernel des Betriebssystems und übergibt Parameter
- Wird in der Regel im MBR platziert (Ausnahme Windows)
- Wird meist über eine Textdatei gesteuert werden
- Bekannte Vertreter unter Linux sind GRUB und bei älteren Systemen LILO



LILLO

- LILLO steht für Linux LOader
- kann Windows und Linux booten
- wird meist auf älteren Systemen verwendet
- Konfiguration über `/etc/lilo.conf`
- hat so gut wie keine Bedeutung mehr



GRUB

- GRUB bedeutet GRand Unified Bootloader
- Dieser universelle Bootloader wird von allen „großen“ Distributionen unterstützt
- Kann neben Windows und Linux auch noch viele weitere BS laden (z.B. Solaris, BSD etc...)
- Chainloader: Bedeutet, dass GRUB andere Bootloader laden kann, ähnlich wie Kettenelemente
- Partitionsänderung kann zum „Totalversagen“ von GRUB führen



GRUB – menu.lst

- enthält die komplette GRUB Konfiguration
- Beispiel:

```
title    Ubuntu 9.04, kernel 2.6.28-11
```

```
uuid    51c3987080989876
```

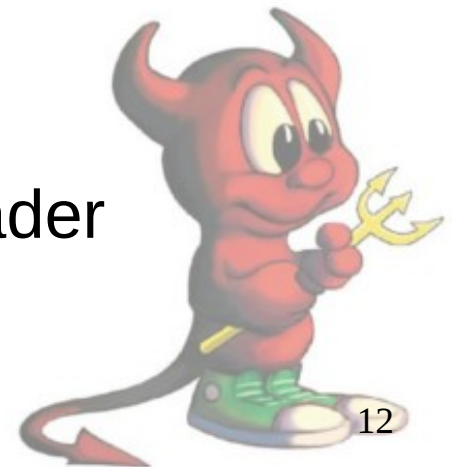
```
kernel  /boot/vmlinuz
```

```
initrd  /boot/initrd.img
```



GRUB – menu.lst Parameter

- default 0 – zu ladendes Standard-BS
- timeout 10 – Zeit bis zum Start des Std.-BS
- title Linux – Titel des BS (kann frei gewählt werden)
- uuid 12-34-56-8 – Festplatten ID
- Kernel /boot/vmlinuz - Kernelimage
- initrd /boot/initrd.img – Image des Initrd
- root (hd0,0) – Partition des zu ladenden BS
- makeactive – Betriebssystem „Aktiv“ machen
- chainloader +1 – Übergibt an den nächsten Loader
- Aufbau steht im Beispiel



GRUB - Installation

- `grub-install (--recheck) hd0` – Installiert Grub im MBR der ersten Festplatte
- `update-grub` – Durchsucht die Festplatte nach bootbaren Partitionen und schreibt sie in den MBR



Start des Kernels

- Prüfen des BIOS Registers und lädt entsprechende Hardwaretreiber
- kswapd wird gestartet
- Rootdateisystem wird gestartet → wenn nicht gefunden → Kernelpanic
- Überprüfung des Hauptspeichers, CPU
- Auswertung der übergebenen Bootparameter



Init.d – und die Runlevel

- /sbin/init – Erster Prozess der vom Kernel initialisiert wird, erhält die PID 1
- Ist der Elternprozess aller anderen Prozesse
- Entstehen durch forking des Init-Prozesses
- Hauptaufgabe:

Initialisieren von Runlevelscripts

- Bei Runlevelscripts handelt es sich um Scripts die beim Start oder Beenden eines Runlevels ausgeführt werden



Runlevel

- Runlevel: Ist ein Systemstatus bei dem bestimmte Dienste laufen
- Entsprechend starten/stoppen Init-Scripts die entsprechende Dienste zu den Runleveln
- In der Regel sind diese Scripte in der Syntax der BASH
- Runlevel stammen aus dem System-V-R4 und finden Verwendung unter Linux/BSD/Solaris



Die Ausprägungen der Runlevel

- 0 – halt: Hält das System an und schaltet es von selbst ab (wenn möglich z.B. ACPI)
- 1 – Single User Modus: nur der „root“ hat Zugriff auf dem Rechner, nur Konsole
- 3 – Multiuser-Modus: Mehrbenutzerbetrieb, nur Konsole
- 5 (je nach Distro) GUI: Mehrbenutzerbetrieb mit GUI
- 6 – Reboot: Neustart des Systems



inittab

- Sagt den Bootloader, welcher Runlevel Standardmäßig geladen werden und welche Aktion ausgeführt werden sollen
- liegt in /etc/inittab (Ausnahme Ubuntu 9.04)
- Wenn init einen Prozess starten soll, wird auf /etc/initscript geprüft, ist dem so, wird der Prozess gestartet
- Die jeweiligen Runlevelscripts liegen in /etc/rc.d/rcX (X ∈ {0,1,2,3,5,6})



inittab - Aufbau

- ID: definiert eine ID, 1 – 4 Zeichen die zur Identifikation der Aktion
- Runlevel: Der für diese Aktion aufgerufen werden soll
- Aktion: legt die Aktion selbst fest, dabei wird ein Schlüsselwort angegeben um Init zu sagen was getan werden soll (z.B. boot, der Prozess wird während des Bootvorgangs ausgeführt).
- Kommando: Welcher Prozess gestartet werden soll

