Alles, was Sie noch nie über das Booten von Unix-Systemen wissen wollten und nach diesem Vortrag sofort wieder vergessen werden.

Sebastian Marius Kirsch
skirsch@moebius.inka.de

10. Mai 2001

Grober Überblick

1. Boot-PROM (aka. BIOS, PROM) lädt Bootloader
2. Bootloader lädt Kernel
3. Kernel mountet root-Dateisystem
4. Kernel startet `/sbin/init`
5. `init` startet weitere Prozesse

Booten von Festplatte

- PROM lädt den ersten Block (512 Byte) der Festplatte in den Speicher und führt ihn aus

- Zu wenig Platz für kompletten Bootloader $\Longrightarrow$ Bootloader wird zweigeteilt

- Ein Teil im Bootblock, der Rest ‘sonstwo’.

(Net/Free)BSD:

- drei-/vierteiliger Bootstrap:
  1. 512 Byte im Bootblock
  2. 5–10KB im Disklabel (kann Dateisystem lesen)
  3. 60KB im root-Verzeichnis
  4. bei FreeBSD: loader verwaltet Kernel-Module und startet den Kernel

Linux mit LILO:

1. 512 Byte im Bootblock

2. lädt `/boot/boot.b` (6 KB, extrem klein!)

   - Blocknummern von Kernel, Boot-Sektoren etc. werden in `/boot/map` gespeichert

   - verloren, sobald sich der Ort des Kernel-Image ändert

Linux mit Grub:

1. 512 Byte im Boot-Sektor

2. optional: lädt stage1.5 aus unbenutztem Bereich in der ersten Spur (5 KB); stage1.5 findet stage2 über das Dateisystem

3. stage2 (70 KB) lädt Konfigurationsdatei und bootet Kernel über das Dateisyste (ext2, ufs, reiserfs, FAT)

Booten von CDROM auf Intel

- El-Torito-Format
- Floppy-Image wird in eine Datei auf der CD geschrieben
- Pointer auf Floppy-Image im Header des ISO9660-Dateisystems
- BIOS lädt Floppy-Image und behandelt es wie ein Diskettenlaufwerk

NB: Auf VAX/VMS wird einfach ein bootbares Dateisystem-Image auf die CD geschrieben.

NB: Auf SPARC wird ein Disklabel auf die CD geschrieben; der Boot-Code für die einzelnen Plattformen wird in verschiedene Partitionen geschrieben.

Booten via Netzwerk

- Verfahren, um die IP-Adresse zu bestimmen:
  - RARP (Reverse Address Resolution Protocol) – einfachste Methode, ermöglicht nur die Bestimmung der IP-Adresse $\Longrightarrow$ Server, der auf RARP-Anfrage antwortet, muss auch weitere Anfragen beantworten. Benutzt von SPARC auf PROM-Ebene.

    Gegenteil von ARP; arbeitet auf der selben Schicht wie ARP/IP.
  - BOOTP (Boot Protocol) – weitergehende Konfigurationsmöglichkeiten wie Dateiname und zuständiger Server, Gateways, DNS-Server etc. Setzt auf IP auf (Port 67/68.)

- DHCP (Dynamic Host Configuration Protocol) – Nachfolger von BOOTP, abwärtskompatibel zu diesem. Komplette Konfiguration des Netzwerkes (samt Font-Servern, Netzwerkdruckern, etc. etc.) Konzept der 'Leases': IP-Adressen werden auf Zeit vergeben. Protokoll der Wahl, benutzt von NetBSD, Linux auf Kernel-Ebene, PXE, X-Terminals.

- Bootparams – benutzt von Solaris, NetBSD/sparc-Bootloader. Basiert (wie z. B. NFS) auf RPC (Remote Procedure Calls).

- Dateitransfer:
  - TFTP (Trivial File Transfer Protocol) – einfachstes Protokoll. Verbindungslos, keine Authentifizierung, geringer Overhead. Benutzt von SPARC auf PROM-Ebene, Grub.
  - NFS (Network File System) – eigentlich Protokoll für File Sharing, wird aber von NetBSD-Bootloader benutzt, um den Kernel zu übertragen. Vorteil: Authentifizierung des Clients, aber erhöhte Komplexität.
  - MOP (Maintenance Operations Protocol) – proprietäres Protokoll von DEC, zur Fernwartung und Booten, benutzt auf VAXen/DECstations. Basiert nicht auf IP, sondern übertragt Datei direkt auf Ethernet-Ebene. *Exot.*

Kernel kann ein String als Kommandozeile übergeben werden

- Linux: reichhaltige Kommandozeile, Grossteil der Hardware kan über Kommandozeilen-Argumente konfiguriert werden. Alle Argumente, die der Kernel nicht kennt, werden an `init` weitergeleitet.

  Wichtige Option: `-b` emergency boot.

- BSD: Spartanisch, erkannte Optionen: `-d` debug, `-a` ask for root, `-s` single-user

Kernel wird gestartet, initialisiert Hardware etc. Mountet dann das Root-Dateisystem.

- Linux: Device-Name des Root-Dateisystems wird direkt im Kernel gespeichert (mit `rdev`) oder auf der Kommandozeile des Kernel übergeben.

- *BSD: fest eincompiliert, oder Bootloader setzt Default-Root-Dateisystem, oder Kernel (bei Boot-Option `-a`) fragt den Benutzer.

- Bei Netzwerk-Boot: Kernel konfiguriert sich über BOOTP/DHCP selbst und mountet Root-Dateisystem über NFS.

`init`

- Vater aller Prozesse, PID 1.
- startet die Systemprozesse
- erbt Prozesse, deren Eltern gestorben sind
- ein System ohne `init` ist nicht tot, aber so gut wie.

SystemV-`init`

- Zentrale Konfigurationsdatei `/etc/inittab`
- verschiedene runlevels (0=shutdown, 6=reboot, andere beliebig, z. B. 1=kein Netzwerk, 2=default, 3=kein X)
- verwendet von Linux und kommerziellen Unices

- Scripts in `/{etc,sbin}/init.d`, die mit den Argumenten `start` und `stop` aufgerufen werden, um Services zu starten/stoppen

- von dort *symbolische* Links nach `/{etc,sbin}/rc[0-6].d/`, die festlegen, in welchem Runlevel welche Dienste gestartet/gestoppt werden

- Dateinamen in `rc[0-6].d`: `[KS][0-9][0-9]*`, `K`: Dienst wird gestoppt, `S`: Dienst wird gestartet, Nummer: Reihenfolge des Startens/Stoppens

BSD-Init

- Verwendet von BSD-Unix, Slackware? (simpleinit)
- keine runlevels
- startet `/etc/rc` beim Systemstart

- lokale Dienste in `/etc/rc.local`
- Bei NetBSD: Skripte in `/etc/rc.d` mit depend/provide-Informationen
- `gettys` in `/etc/gettytab`

Links

- GNU GRUB
  http://www.gnu.org/software/grub

- The FreeBSD Booting Process
  http://www.freebsd.org/doc/en_US.ISO_8859-1/books/handbook/index.html

- NetBSD: HOWTO Netboot a diskless machine
  http://www.netbsd.org/Documentation/network/netboot/

- NetBSD/VAX Netboot HOWTO
  http://world.std.com/~bdc/projects/vaxen/VAX-netboot-HOWTO.html

- Linux BootPrompt HOWTO
  http://www.linuxdoc.org/HOWTO/BootPrompt-HOWTO.html

- Linux Diskless HOWTO
  http://www.linuxdoc.org/HOWTO/Diskless-HOWTO.html

- Sebastian Marius Kirsch: *Larve Nimmersatt*, Mehrere Betriebssysteme mit dem GNU-Bootloader Grub
  http://www.heise.de/kiosk/archiv/ct/00/20/238/