

Systemadministration Teil 8

Prof. Dr.-Ing. Jörn Schneider

Wiederholung

Boot-Vorgang

- Beim Einschalten: HW sorgt für Reset am Prozessor (hard Reset)
- Fetch von Startadresse (Typischerweise Adresse 0x0)
- Hier ist NV-RAM (früher ROM, heute Flash) eingeblendet von dem aus die Firmware ausgeführt wird
 - NV-RAM = Non-Volatile Random Access Memory
 - Bei PCs bezeichnet man die Firmware als BIOS
- Die Firmware lokalisiert den Datenträger von dem das zu startende Bootprogramm geladen werden soll
 - Weitere wichtige Schritte der Firmware
 - Durchführen HW-Test, POST (Power-on-self-test)
 - Konfiguration der HW

Boot-Vorgang (2)

- Die Firmware lädt das Bootprogramm aus dem MBR (Master Boot Record) in den Speicher und führt es aus
- Das Bootprogramm identifiziert die aktive Partition und lädt von dort den Betriebssystemkernel
 - Das Bootprogramm kann insbesondere auch ein Bootmanager sein, bzw. vor dem Betriebssystemkernel einen solchen laden

Boot unter Linux

1. Evtl. Ausführung Bootmanager
 - LILO
 - GRUB
 - U-Boot
2. Der Kernel wird geladen und ausgeführt
3. Kernel initialisiert die Hardware und lädt die notwendigen Treiber
 - autoconfigure, d.h. ansprechen aller möglicherweise vorhandenen Hardware und Registrierung der antwortenden Geräte
4. Prozess 0 wird gestartet
5. Prozess 0 mounted das root File System und erzeugt die Prozesse 1 (init) und 2 (page daemon)

Boot unter Linux (2)

6. Der *Init* Prozess ist Stammutter/-vater aller weiteren Prozesse
 7. Weitere Initialisierungen (siehe Initphase)
 8. Normalbetrieb: Start von *getty* für jedes Terminal
 - TTY kommt von Teletype Writer (Fernschreiber)
 - *getty* konfiguriert Terminal, schreibt "login:" und wartet auf Eingabe
-
9. Bei Eingabe Benutzername terminiert *getty* durch Start von *login*
 10. *login* fragt nach Passwort, verschlüsselt dieses und vergleicht mit Eintrag in */etc/shadow*
 11. Nach erfolgreichem Anmelden terminiert *login* durch Ausführung der Shell des Benutzers

Was passiert in Initphase?

- Überprüfen des Systems
 - Filesystem Check
- Einrichten der benötigten Umgebung
 - Mount der benötigten Filesysteme
 - Netzwerkanbindung
- Starten der jeweils benötigten Dienste (Dämonen) in der korrekten Reihenfolge, z.B.:
 - Druckerdienst (Printer Dämon)
 - Mailserver

Wie wird die korrekte Abfolge eingehalten?

- Es werden verschiedene sogenannte Runlevel durchlaufen
- Für den Normalfall ist die Abfolge der Runlevel beim Bootvorgang per Konfiguration festgelegt.
- Im Betrieb kann der Systemadministrator in andere Runlevel wechseln
- Auch das Herunterfahren des Systems wird über spezielle zu durchlaufende Runlevel realisiert

Ende Wiederholung

Runlevel

- S: Start
- 0: Halt
- 1: Single User („sauber“)
- 2: Default Multi User, d.h. Netzwerkdienste gestartet
- 3-5: Weitere Multi User Runlevel
- 6: Reboot

Beispiel Sequenz Runlevel

- Normaler Start:
- 1. runlevel S
- 2. runlevel 2
- Sysadmin schaltet in single user: init 1
- 3. runlevel 1
- Sysadmin schaltet zurück in multi user: init 2
- 4. runlevel 2

Wiederholung

Realisierung Runlevel Skripte

- rc-Skripte (in /etc/rc?.d) sind in Wirklichkeit symbolische Links auf Skripte im Verzeichnis /etc/init.d
- Dort gibt es für jeden Dienst **ein** Konfigurationsskript
- Link beginnt mit K → Aufruf mit Parameter „ stop“
- Link beginnt mit S → Aufruf mit Parameter „ start“

Fingerübung: Runlevel

- Im Verzeichnis `/etc/rcS.d` existieren folgende Einträge:
S01Bdienst, S14Adienst
- Im Verzeichnis `/etc/rc1.d` stehen folgende Einträge:
S01Cdienst, K99Ddienst
- Im Verzeichnis `/etc/rc2.d` stehen folgende Einträge:
S01Ddienst
- Das System läuft im default runlevel (2).
- Welche Skripte werden in welcher Reihenfolge gestartet, wenn der Systemadministrator mit dem Kommando "init 1" in den Single User Modus schaltet und welche Parameter werden dabei jeweils übergeben?

Ende Wiederholung

BEISPIELE FÜR DÄMONEN

CRON DAEMON

cron Daemon

- Aufgabe - regelmäßige Ausführung von Aktivitäten (cron Jobs) in festen Zeitintervallen
 - z.B.
 - Jede Minute prüfen, ob Mails eingegangen sind
 - tägliches Backup durchführen (inkrementell)
 - wöchentliches Backup durchführen (vollständig)

VERWALTUNG VON CRON JOBS

cron Jobs ...

- Werden zu den festgelegten Zeiten vom cron Daemon gestartet
- Laufen im Hintergrund
- Können also nicht interaktiv sein
 - /dev/null wird nach stdin umgeleitet
- Eventuelle Ausgabe wird per Mail an entsprechenden User gesendet

System cron Jobs

/etc/crontab

- Tabelle mit System cron Jobs

Format:

- <Zeit> <Kommando>
- Zeit=
<Min> <Stunden> <Tag des Monats> <Monat> <Tag der Woche>
- Kommando=
<ausführbare Datei> [<Parameter>] [%<Text für stdin>]

Beispiele

- 0,30 * * * * write notroot %,,Wieder eine 1/2 Stunde rum,,
- 30 10 * * 1 write notroot %,,Manic Monday,,

Fingerübung: Cron Jobs

User cron Jobs

- Kommando crontab
 - Anzeigen
 - `crontab -l`
 - Einrichten/Ändern/Löschen
 - `VISUAL=`which vi`; export VISUAL`
 - `crontab -e`
- Files sind abgelegt in z.B. `/var/spool/cron/crontabs/`

CRON DAEMON START UND STOP

Start des cron Daemons

- In allen Multiuser Runlevel
- z.B. `/etc/rc2.d/S89cron`
 - bewirkt Ausführung von `"/etc/rc2.d/S89cron start"` bei Wechsel in Runlevel 2
 - S89cron ist symbolischer link auf `/etc/init.d/cron`

Stop des cron Daemons

- In single user Runlevel (1)
- z.B. `/etc/rc1.d/K11cron`
 - bewirkt Ausführung von `"/etc/rc1.d/K11cron stop,` bei Wechsel in Runlevel 1
 - `K11cron` ist symbolischer link auf `/etc/init.d/cron`