

Systemstart und Systemstopp

Lernziele



- Die Vorgänge beim Systemstart verstehen
- Runlevels und Systemstart konfigurieren
- Die Vorgänge beim Herunterfahren des Systems durchblicken

Der Bootvorgang

- Nach dem Einschalten des Rechners bekommt das BIOS die Kontrolle
- Die Hardware wird getestet und initialisiert
 - Memory
 - Speichermedien (Harddisk, CD/DVD, Floppy)
 - Tastatur, VGA
 - Schnittstellen (Paralell, Seriell, USB)
- Erweiterungshardware wird initialisiert
 - Diskkontroller (SCSI, RAID)
 - Netzwerkkarten (PXE)

BIOS, Bootlader und Kernel

- Anschliessend wird anhand der im BIOS eingestellten Bootreihenfolge nach einem Bootlader gesucht
- Der Bootlader wird in den Hauptspeicher kopiert und übernimmt die Kontrolle
- Es werden weitere Teile des Bootmanagers, z.B. ein Menü geladen
- Dann wird der Betriebssystemkern ins RAM geladen und gestartet

BIOS, Bootlader und Kernel

- Der Kernel initialisiert die Hardware für Linux
 - Auf Intel-i386 Prozessoren wird nun vom *real Mode* in den *Protected Mode* geschaltet
 - 32-Bit oder 64-Bit Modus
 - Das BIOS ist nun nicht mehr zugänglich
 - Es wird ein *Paging-System* bereit gestellt
 - Hardwareunterstützung für virtuellen Speicher

BIOS, Bootlader und Kernel

- Nun kann die eigentliche Arbeit des Kernels anfangen
- Initialisierung des gesamten virtuellen Speichers
- Aufbau der Interrupt und Service-Routinen
- Parameterisierung des Schedulers
- Der init-Prozess wird gestartet
- Er nimmt die weiteren Initialisierungen ausserhalb des Systemkerns vor

Der Init-Prozess

- Der Init-Prozess hat die Nummer 1
- Er kann nicht durch ein Signal (z.B. *kill -9*) beendet werden
- Aufgaben des Init-Prozesses sind:
 - Weiterführen des Bootvorgangs
 - Verwaltung der Runlevels
 - Anmeldevorgang auf virtuellen Konsolen
- Die Vorgaben des Init-Prozesses stehen in */etc/inittab*

Die Datei /etc/inittab

```
# Standard-Runlevel
id:5:initdefault

# Erstes auszuführendes Skript
si::bootwait:/etc/init.d/boot

# Runlevels
l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
#l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6
```

Die Datei /etc/inittab

(Fortsetzung)

```
ls:S:wait:/etc/init.d/rc S
~~:S:respawn:/sbin/sulogin
```

```
# Ctrl-Alt-Del
```

```
ca::ctrlaltdel:/sbin/shutdown -r -t 4 now
```

```
# Terminals
```

```
1:2345:respawn:/sbin/mingetty --noclear tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

Der Init-Prozess

- Der Init-Prozess startet ein Shellsript, das **Bootscript**
 - `/etc/init.d/boot` (SUSE)
 - `/etc/rc.d/init.d/boot` (Red Hat)
- Das Bootsript übernimmt Aufgaben wie
 - die Fehlerprüfung und etwaige Korrektur der in `/etc/fstab` eingetragenen Dateisysteme
 - die Initialisierung des Rechnernamens
 - das Einstellen der System-Uhr
 - mounten der Dateisysteme

Der Init-Prozess

- Die spezifischen Aktionen und ihre genaue Reihenfolge hängen von der verwendeten Linux-Distribution ab
- Die verschiedenen Dateien in */etc/init.d/boot.d* werden ausgeführt
- Hier kann der Administrator weitere Shell-Scripte anlegen
- Nach dem Bootscript werden die Runlevel gestartet

Runlevel

- 0 System-Halt
- 1 bzw. S Einbenutzermodus ohne Netzwerk
- 2 Mehrbenutzermodus ohne Netzwerk
- 3 Mehrbenutzermodus mit Netzwerk
- 4 Unbenutzt, kann individuell konfiguriert werden
- 5 Mehrbenutzermodus mit Netzwerk und grafischer Anmeldung
- 6 Reboot

Runlevel

- Beim Systemstart wird in der Regel in den Runlevel 3 oder 5 gestartet
- Runlevel 3 ist typisch für Server Systeme
 - keine grafische Oberfläche
 - evtl. keine Grafikkarte vorhanden
- Runlevel 5 ist für Arbeitsplatzsysteme
 - X-Server wird lokal gestartet
 - Window Manager
 - Grafisches Login



Runlevel

- Während des Betriebs kann der Runlevel mit dem Befehl *init* gewechselt werden
 - # `init <Runlevel>`
- Im neuen Runlevel nicht mehr benötigte Dienste werden beendet
- Im neuen Runlevel erforderliche Dienste werden gestartet

Runlevel

- Mit dem Kommando *runlevel* können der vorige und der aktuelle Runlevel abgefragt werden
- # runlevel
N 5
- Hier befindet sich das System im Runlevel 5
- Der vorige Runlevel ist “N“
- Das System wurde direkt in den Runlevel 5 gebootet

Runlevel

- Die Dienste, in den verschiedenen Runlevels, werden über **Init-Skripte** gestartet und gestoppt
 - /etc/init.d (SUSE)
 - /etc/rc.d/init.d (Red Hat)
- Die Skripte werden beim Wechsel von einem Runlevel in den anderen ausgeführt
- Können auch von Hand ausgeführt werden
- Es können eigene Skripte eingefügt werden

Runlevel

- Die Init-Skripte werden über Parameter gesteuert
- *start* wird im neuen Runlevel gestartet
- *stop* wird im neuen Runlevel beendet
- *status* Abfrage, ob der Dienst läuft
- *restart* der Dienst wird beendet und neu gestartet
- *reload* der Dienst liest seine Konfiguration neu ein

Runlevel

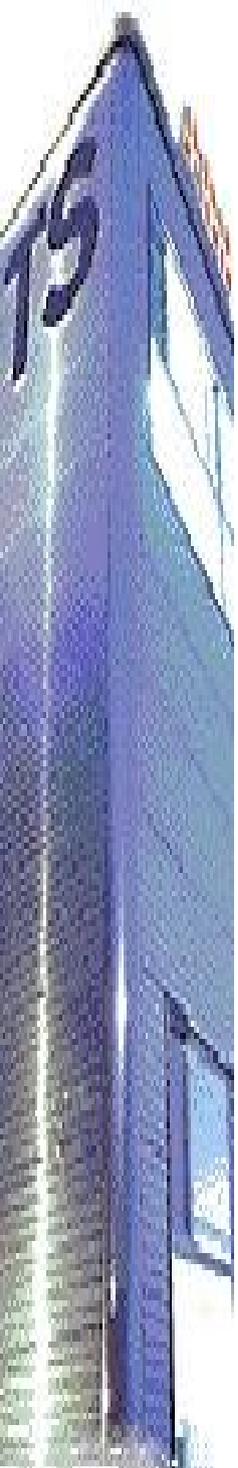
- Zu jedem Runlevel gibt es ein Unterverzeichnis
 - `rc<r>.d`
- In den Verzeichnissen befinden sich symbolische Links zu den Skripten
- Über diese Links werden die betreffenden Skripte gestartet oder gestoppt von
 - `/etc/init.d/rc`

Runlevel

- Beim Start eines Runlevel werden alle symbolischen Links, die mit "S" beginnen mit dem Parameter "*start*" aufgerufen
- Beim Beenden eines Runlevel werden die symbolischen Links, die mit "K" beginnen mit dem Parameter "*stop*" aufgerufen
- Die Namen der symbolischen Links werden in lexikographischer Reihenfolge aufgerufen

Runlevel

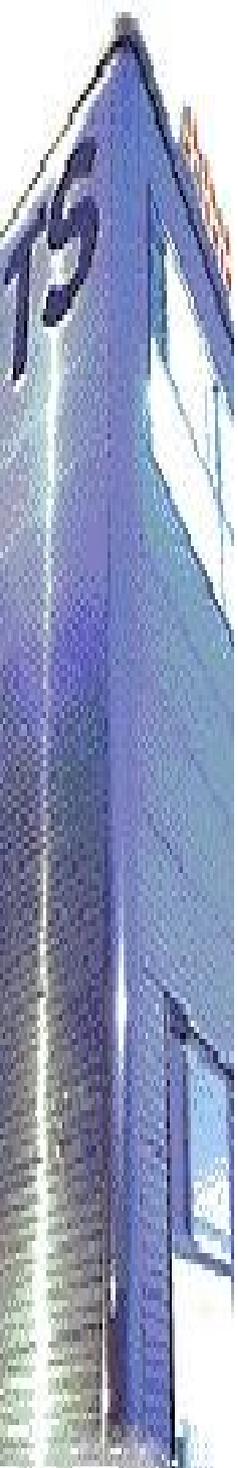
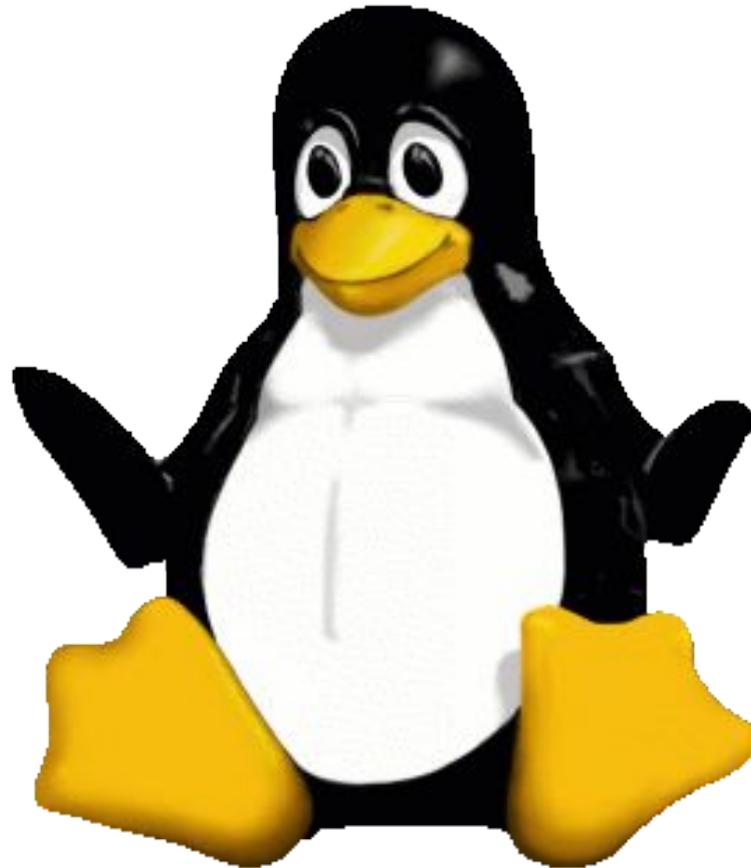
- Die Namen der Links enthalten nach dem “S” eine zweistellige Zahl
- Dadurch kann bestimmt werden, wann im Laufe dieses Vorgangs welcher Dienst gestartet wird
- Zum Deaktivieren der Dienste für einen Runlevel wird ebenfalls eine zweistellige Zahl nach dem “K” verwendet



Runlevel

- Dienste konfigurieren kann man prinzipiell durch anpassen der symbolischen Links
- Die SUSE-Distributionen verwendet
 - # `insserv`
 - Es benutzt Informationen in den einzelnen Init-Skripten
- Zum Konfigurieren der Runlevel dient
 - # `chkconfig`
 - # `chkconfig -l`
 - Zeigt die konfigurierten Runlevel

Fragen?

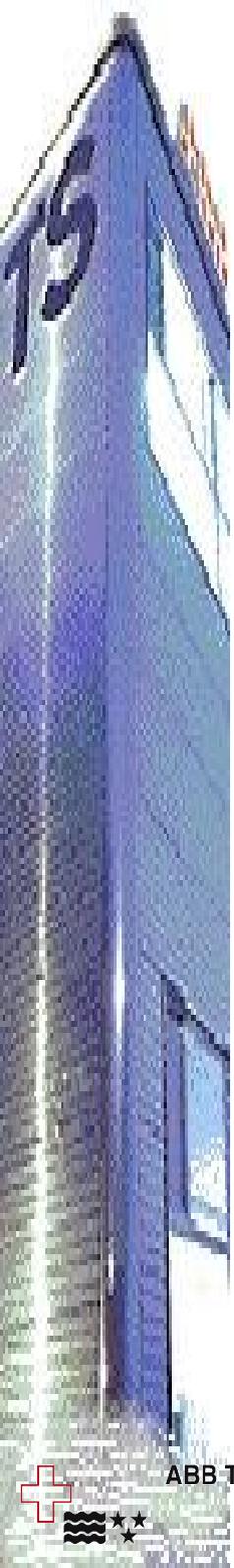


Übungen

- [6.1] Lassen Sie sich den momentanen Runlevel anzeigen.
- Was genau wird angezeigt?
- Wechseln Sie in Runlevel 2
- Sehen Sie sich jetzt den aktuellen Runlevel an.

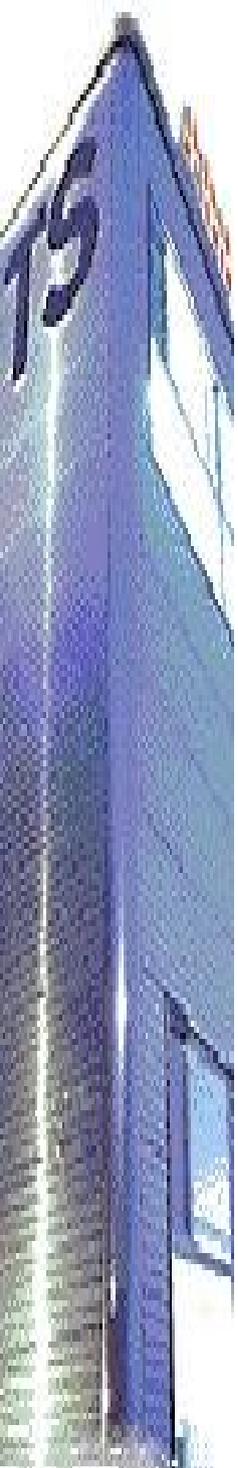
Übungen

- [6.2] Was müssen Sie tun, damit der syslog-Dienst seine Konfiguration neu einliest?



Übungen

- [6.3] Wie können Sie bequem die momentane Runlevel-Konfiguration überprüfen?



Übungen

- [6.4] Entfernen Sie den Dienst *cron* aus Runlevel 2.



Der Einbenutzermodus

- Im Einbenutzermodus (Runlevel 1 oder S) kann nur der Systemadministrator an der Konsole arbeiten
- ! Es gibt nur eine einzige Konsole
- Wird für Administrationsaufgaben und Reparatur am Dateisystem verwendet
- Zum Starten des Single-user-mode muss am Bootprompt die Option "1" oder "S" angegeben werden

Der Einbenutzermodus

- Das Rootfilesystem kann z.B. beim Booten nur zum Lesen gemountet werden
- Man kann die Schreibberechtigung auch »von Hand« ändern
 - `# mount -o remount,ro /`
- Es dürfen keine Prozesse eine Datei im Dateisystem zum Schreiben geöffnet haben
- In der Regel Daemons wie *syslogd*, *lpd*, *cron*

Herunterfahren des Systems

- Warum herunterfahren?
 - Die verschiedenen Dienste sollten beim Abschalten des Systems kontrolliert beendet werden
 - Linux nutzt einen Teil des Arbeitsspeichers als Zwischenspeicher. Dieser muss noch auf die Platte geschrieben werden
 - Kann mit dem Befehl `sync` forciert werden
 - Plötzliches Ausschalten kann zu Datenverlusten und im schlimmsten Fall zur Zerstörung des Dateisystems führen

Herunterfahren des Systems

- Es gibt verschiedene Möglichkeiten, das System anzuhalten:
 - Mit dem Befehl *shutdown*
 - Dieses Kommando ist die sauberste Form, das System herunterzufahren
 - Mit der Tastenkombination »Alt+Strg+Entf«
 - Mit dem Befehl »*init 0*« wird der Runlevel 0 angefordert
 - Mit dem Kommando *halt*

Das shutdown-Kommando

- Das Programm *shutdown* sorgt dafür, dass alle Benutzer und Prozesse über die bevorstehende Aktion informiert werden

- `shutdown -h +10`

- 1 Broadcast-Message an alle Benutzer
 - 2 Die Datei `/etc/nologin` wird angelegt
 - 3 Wechsel in den Runlevel 0
- Alle Dienste werden über ihre Init-Skripte beendet

Das shutdown-Kommando

(Fortsetzung)

- 4 Alle noch laufenden Prozesse erhalten zuerst das Signal SIGTERM
- 5 Kurze Zeit später werden alle bis dahin nicht beendeten Prozesse durch das Signal SIGKILL beendet
- 6 Die Dateisysteme werden ausgehängt und der Swapbereich deaktiviert
- 7 Alle Aktivitäten des Systems werden beendet

Übungen

- [6.5] Fahren Sie Ihr System von jetzt an in fünfzehn Minuten herunter und teilen Sie den Benutzern mit, dass das Ganze nur ein Test ist.
- Wie verhindern Sie dann den tatsächlichen Shutdown?
(damit das ganze wirklich nur ein Test ist)

Der Befehl sync

- Linux unterhält zur Optimierung der Festplatten und Diskettenzugriffe ein Blockdepot (Cache)
- Veränderungen an diesen Daten werden zuerst nur im Arbeitsspeicher aktualisiert
- Die veränderten Datenblöcke werden regelmässig vom Update-Daemon auf die Festplatte zurückgeschrieben
- Mit `sync` kann man das gezielt ausführen

Neustart

- Ein Neustart ist notwendig, wenn bestimmte Hardware neu installiert oder initialisiert wird
- Partitionstabelle, initrd, Map-Dateien
- Wenn sich ein Systemabsturz ankündigt
- Einen Neustart auslösen kann:
 - `shutdown -r now`
 - `init 6`
 - `reboot`