



www.lpi.org



 Linux Professional Institute

- ✓ Vendor Independent
- ✓ Community Supported
- ✓ Respected Worldwide



Inhalt

101.2 Booten des Systems

101.3 Ändern des Runlevels, Shutdown, Neustart des Systems



101.2 Booten des Systems

Beschreibung: Prüfungskandidaten sollten in der Lage sein, das System durch den Bootprozess zu führen. Dieses Lernziel beinhaltet das Übergeben von Kommandos an den Bootlader und die Übergabe von Optionen an den Kernel zum Bootzeitpunkt sowie das Überprüfen von Ereignissen in den Logdateien.

Die wichtigsten Dateien, Bezeichnungen und Anwendungen:

- /var/log/messages
- dmesg
- bootloader
- init
- kernel



Der Bootvorgang von Linux

Die Funktion eines Bootmanagers wurde bereits behandelt und wird deshalb vorausgesetzt.

Start von Systemdiensten und Servern

Systemstartkomponente	Konfiguration
Rechner (i386-Plattform): Grafikkarten-BIOS und Rechner BIOS wird geladen	
Bootlader im MBR lädt Kernel und initrd	grub: /etc/grub.conf /boot/grub/menu.lst (man grub)
Hardwareinitialisierung durch den Kernel	Kernelkompilierung, Kernelparameter Über- gabe beim Start
mounten von / (read only)	
Start des init-Prozesses	/etc/inittab



Der Bootvorgang von Linux (2)

Start von Systemdiensten und Servern

Systemstartkomponente	Konfiguration
Start des runlevel unabhängigen Bootscripts	/etc/init.d/boot SuSE /etc/init.d/rcS Debian
Aufrufen des runlevel abhängigen Bootscripts	/etc/inittab bzw. Übergabe des gewünschten Runlevels am Bootlader Prompt
Runlevel-Scripts starten und stoppen Systemdienste (Daemons) und Server	/etc/rc* setzen von Sym-links auf die Srippte in /etc/init.d



Kernelparameter übergeben

Der Linux-Kernel kann beim Start Parameter übergeben bekommen, die sich auf die Art und Weise auswirken, wie er startet bzw. wie er sich zur Laufzeit verhalten soll.

- Parameter für bestimmte Geräte handeln (etwa SCSI-Adressen des Bootlaufwerks)
- Eigenschaften des Kernels selbst (z.B. `acpi=no`)
- Startverhalten ändern (`init=1` oder `init=/bin/bash`)

Kernelparameter für Geräte werden nur notwendig, wenn mit Hardware gearbeitet wird, die der Kernel nicht selbstständig erkennt. Ein Standard-PC mit Standard-Komponenten benötigt keine zusätzlichen Kernelparameter.

Regel: Alle für einen Treiber relevanten Parameter müssen unmittelbar hintereinander, durch Kommas getrennt, angegeben werden. Es darf kein Leerzeichen zwischen den einzelnen Parametern eines Treibers sein. Sollen mehrere Kernelparameter angegeben werden, so werden diese mit Leerzeichen voneinander getrennt. Z.B.

`aha152x=0x300,10,7`



Kernelparameter (2)

Auf dem Bootprompt von LILO oder GRUB

Der Bootmanager erwartet die Nennung des zu startenden Systemnamens (z.B. des in /etc/lilo.conf angegebenen Labels). Diesem Namen können Parameter mitgegeben werden. z.B.:

LILO boot: linux aha152x=0x300,10,7

Dieser Kernel-Parameter sagt dem Kernel, daß ein Adaptec AHA152X SCSI Hostadapter an der Adresse 0x300 sitzt, den IRQ 10 benutzt und die SCSI-ID 7 belegt. Analog auch auf der Boot-Zeile von GRUB). Diese Form der Parameterübergabe ist ideal für experimentelles booten. Durch Ausprobieren einzelner Parameter kann man sicherstellen, dass sie funktionieren.

In der Datei /etc/lilo.conf

Sollen diese Parameter dauerhaft eingerichtet werden, so ist die Datei /etc/lilo.conf der richtige Ort (sowohl in der globalen ersten Sektion, als auch für jeden Kerneleintrag einzeln mit dem Schlüsselwort `append=`) diese anzugeben.

In der Datei /boot/grub/grub.conf

Hier werden die Parameter direkt an die Kernelangabe gehängt, also etwa in der Art

kernel /bzImage-2.6.18 ro root=/dev/hda3 aha152x=0x300,10,7

In der Datei /etc/conf.modules bzw. modules.conf

Nachträglich ladbare Module können in der Datei /etc/conf.modules konfiguriert werden. Auch hier geht es dann um Kernel-Parameter für ladbare Module.



Kernelmodule konfigurieren

Die Vorteile eines modularen Kernels liegen in seiner Flexibilität und der Möglichkeit Hardware-Treiber genau dann zu laden, wenn sie gebraucht werden.

Die notwendigen Angaben eines Treibers werden in der Datei **/etc/conf.modules** oder **/etc/modules.conf** eingetragen. (Meist ist modules.conf ein Link auf conf.modules oder umgekehrt).

[Manche Kernel-Module für ältere ISA-Karten benötigen Angaben über ihre verwendeten Adressen, IRQs, DMA-Kanäle u.a.]

```
alias eth0 8139too
```

```
(1. Ethernetkarte, modul 8139too)
```

```
alias usb-controller usb-uhci
```




Meldungen des Bootvorgangs nachvollziehen

Wie der **syslogd** schreibt auch der **klogd** (Kernel Log Daemon) Meldungen in die Datei `/var/log/messages`.

Der Kernel schreibt alle Meldungen, die beim Start ausgegeben werden in einem Buffer, der mit dem Programm **dmesg** angezeigt werden kann. (Alle Meldungen des Kernels, die er seit dem Booten erstellt hat, bzw. wenn der Buffer voll ist, die letzten Meldungen, die in den Buffer passen).

Damit stehen auch die Kernel-Meldungen zur Verfügung, die vom Syslog-Daemon bzw. Klog-Daemon noch nicht protokolliert werden konnten, weil diese Daemons noch gar nicht geladen waren als die Meldungen anfielen.

```
dmesg | less
```

```
dmesg > bootmsg.txt
```

Meldungen können gelesen oder in eine Datei geschrieben werden.



101.3 Ändern des Runlevels, Shutdown und Reboot

Beschreibung: Die Kandidaten sollen in der Lage sein, die Runlevel des Systems zu verwalten. Dieses Lernziel beinhaltet auch den Wechsel in den Single-User-Modus, das Herunterfahren oder den Neustart des Systems. Die Kandidaten sollen in der Lage sein, andere Benutzer vor dem Wechsel des Runlevels zu benachrichtigen und Prozesse ordentlich zu beenden. Dieses Lernziel beinhaltet ebenfalls auch die Festlegung des Standard-Runlevel.

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- /etc/inittab
- shutdown
- Init
- /etc/init.d
- telinit



Der Init-Prozess

Der Init-Prozess (Prozess Nr. 1) arbeitet den Code von **/sbin/init** ab.

Seine Hauptaufgaben sind: die Weiterführung des Bootvorgangs, die Verwaltung der Runlevel, die beschreiben, in welchem Zustand sich das System befindet und welche Dienste es anbietet.

Außerdem kümmert der Init-Prozess sich darum, dass auf virtuellen Konsolen, direkt seriell angeschlossenen Terminals u. ä. ein Anmelden möglich ist, und verwaltet den Zugriff auf das System über etwaige Modems. All dies wird in der Datei `/etc/inittab` konfiguriert. (nächste Folie)

Anschließend startet der Init-Prozess das Bootskript (Shell-Script):

```
/etc/init.d/boot (SUSE)  
/etc/rc.d/init.d/boot (Red Hat)  
/etc/rc.d/rcS (Debian)
```

[Bootskript: z.B. Fehlerprüfung der Dateisysteme, die Initialisierung des Rechnernamens, ... Kernelmodule werden nachgeladen, Dateisysteme gemountet]



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
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
# Seriellles Terminal
# S0:12345:respawn:/sbin/agetty -L 9600 ttyS0 vt102
# Modem
# mo:235:respawn:/usr/sbin/mgetty -s 38400 modem
```



Runlevel

Die verschiedenen Runlevels und ihre Bedeutung sind inzwischen in den meisten Distributionen standardisiert, etwa nach folgendem Schema:

1,s,S Einbenutzermodus ohne Netzwerk

2 Mehrbenutzermodus ohne Netzwerkserver (Debian default)

3 Mehrbenutzermodus mit Netzwerkservern

4 (Unbenutzt)

5 Wie Runlevel 3, aber mit grafischer Anmeldung

6 Reboot

0 System-Halt

- Beim Systemstart wird in der Regel einer der Runlevels (2 Debian) 3 oder 5 gestartet.
- Runlevel 5 ist typisch für Arbeitsplatzsysteme, die eine grafische Umgebung anbieten, während Runlevel 3 für Server sinnvoll ist, die möglicherweise gar nicht über eine Grafikkarte verfügen.

Diese Runlevel-Vorgaben entstammen dem LSB-Standard. Nicht alle Distributionen setzen sie tatsächlich um; Debian GNU/Linux zum Beispiel überlässt die Runlevel-Zuordnung weitgehend dem lokalen Administrator.



Ändern des Runlevels

Nur root kann das aktuelle Runlevel zu ändern. Man gibt auf der Konsole ein:

```
init <runlevel-nr>
```

Anstelle von **init** kann auch das Programm **telinit** benutzt werden, das gleich funktioniert. (telinit ist nur ein SymLink auf init, aus Kompatibilitätsgründen)

Damit wechselt man in das gewünschte Runlevel.

```
runlevel
```

als Befehl gibt das aktuelle Runlevel und das vorherige aus.

```
N 5
```

(N = none, also Neustart, danach Runlevel 5 aufgerufen)

```
3 5
```

(Voriges Runlevel war 3, jetzt befindet sich das System in Runlevel 5)



Start und Stoppskripte

/etc/init.d

Hier findet man alle relevanten Start- und Stoppskripte des Systems. Man kann die einzelnen Dienste von hier aus auch direkt starten.

```
/etc/init.d/httpd stop
```

```
/etc/init.d/httpd start
```

```
/etc/init.d/httpd restart
```

Verzeichnisse rc0.d rc6.d (für die jeweiligen Runlevels)

Hierin sind die Start-/Stoppskripte aus /etc/init.d per Symlinks angelegt.

Sie besitzen K und S Präfixe [S= Startskript, K= Killskript] und eine 2stellige Nummer und den Namen des Dienstes für den sie stehen.

Beispiel: S10network, S12syslog usw.



Einbenutzermodus (Runlevel S oder 1)

- Dieser Einbenutzermodus wird für Administrationsaufgaben aktiviert bei denen mehrere Benutzer stören würden, z.B. wenn das Filesystem repariert werden muß oder eine Platte gespiegelt werden soll ... u.s.w
- Kein Konsolenwechsel mit [ALT] + n möglich. Es steht nur eine Konsole zur Verfügung. (S = Single User Mode)



Herunterfahren eines Systems

Verschiedene Aufgaben sind zu erledigen, um ein System sauber herunterzufahren.

- Korrektes Beenden laufender Dienste und Benutzerprogramme
- die Synchronisation der Plattenlaufwerke (Cache auf Platte schreiben)
- Benutzer benachrichtigen

Varianten:

- **init 0** oder **init 6** (Funktionieren, aber Benutzer werden nicht informiert!)
- **reboot**, **poweroff**, **suspend** und **halt** dienen ebenfalls dazu, das System herunterzufahren (reboot, poweroff und suspend sind Links auf halt)
- Das Programm **shutdown** ist der beste und sicherste Weg, ein System herunterzufahren oder neu zu starten.

Alle angemeldeten Benutzer erhalten eine Nachricht, die sie auf das Herunterfahren hinweist. Das (Neu-)Anmelden wird in blockiert. Es ist möglich, das System sofort herunterzufahren, oder zu einer bestimmten Zeit.



shutdown

Befehl: shutdown [Optionen] Zeit

`shutdown -r`

Reboot. Shutdown fährt das System herunter und startet es dann erneut.

`shutdown -h`

Halt. Shutdown fährt das System nur herunter ohne einen Neustart danach.

`shutdown -c`

Cancel. Ein schon laufender Shutdown wird unterbrochen und somit das System nicht heruntergefahren.

Zeitangaben:

`hh:mm` Die Warnung für die User erscheint aber sofort nach dem Befehlsaufruf und niemand kann sich mehr anmelden.

`+m`

Die Anzahl der Minuten ist, bis das System heruntergefahren wird. Das Wort `now` wird als `+0m` interpretiert.

`shutdown -h now`



Übungen

- (1) Nutzen Sie dmesg und less um die Boot-Messages zu lesen
- (2) Bringen Sie ihr System in den Single User Mode
- (3) Welche Dienste laufen noch?
- (4) Bringen Sie ihr System wieder in den normalen (default Runlevel) Modus
- (5) Machen Sie sich mit 2 Skripten in /etc/init.d vertraut.
- (6) Schauen Sie sich die Symlinks der Skripte in den verschiedenen Runlevels an.
- (7) Welche Möglichkeiten kennen Sie um ihr System zu stoppen?
- (8) Beenden Sie ihr System in 15 Minuten so, dass die Benutzer informiert werden.
- (9) Lassen Sie sich den momentanen Runlevel anzeigen
- (10) Was müssen Sie tun, damit der syslog-Dienst seine Konfiguration neu einliest?
- (11) Booten Sie direkt in Runlevel 1 oder S
- (12) Wie verhindert der Befehl `shutdown -h +10`, dass sich Benutzer am System anmelden können?



Quellen

- (1) Linux Systemadministration, Thomas Erker, Anselm Lingnau, Linup Front GmbH 2005
- (2) LPIC Level1 Study-Guides, F.Kalhammer, 2002
- (3) LPI Linux Certification in an Nutshell, S. Pritchard u.a, O'Reilly 2006