

1.3 Der Bootmanager GRUB 2

ist eine von Grund auf neu geschriebene Version des GRUB. Der Quellcode wurde grundlegend umstrukturiert und aufgeräumt, die Konfiguration wurde komplexer und es kamen einige Features hinzu. Die wichtigsten Unterschiede zu *GRUB 1* alias *GRUB Legacy* sind:

- GRUB 2 wird aktiv weiterentwickelt, während die Entwicklungsarbeit an GRUB 1 eingestellt wurde. Die wichtigsten großen Distributionen setzen mittlerweile standardmäßig GRUB 2 ein (ab Fedora 16, Ubuntu 9.10, Debian Squeeze)
- Für den Benutzer hat sich oberflächlich nichts geändert. Das Bootmenü ist fast identisch mit GRUB.
- Die Konfiguration für den Administrator ist mächtiger aber auch komplexer geworden und ähnelt mittlerweile einer vollwertigen Skriptsprache. Die Konfigurationsdatei wird überdies von Shell-Skripten generiert.

Das hat zur Auswirkung, dass nun wieder nach jeder Konfigurationsänderung ein Update-Befehl ausgeführt werden muss, um die Konfigurationsänderungen wirksam werden zu lassen.

- Partitionsnummern werden nun beginnend von der 1 an nummeriert, statt, wie bisher, von 0.
- GRUB 2 kann nun sowohl direkt von LVM und RAID, als auch von modernen Dateisystemen wie ext4, HFS+ und NTFS booten. Darüber hinaus kann er nun auf unterschiedlichen Systemarchitekturen (EFI, coreboot, PowerPC, SPARC, MIPS) eingesetzt werden.

1.3.1 Überblick und Konfiguration

Die Hauptkonfigurationsdatei von GRUB 2 ist `/boot/grub/grub.cfg` (RedHat: `/boot/grub/grub2.cfg`). Sie sollte jedoch nicht von Hand verändert werden, da sie generiert wird.


Will man die GRUB 2-Konfiguration verändern, so geht man wie folgt vor:

GRUB 2 konfigurieren



1. Editiere `/etc/default/grub`. Die wichtigsten Konfigurationsvariablen sind:

GRUB_DEFAULT Der Eintrag, der gestartet wird, wenn der Benutzer nach einer bestimmten Zeit keine Auswahl trifft.

GRUB_HIDDEN_TIMEOUT Legt die Zeit fest, die GRUB bei leerem Bildschirm wartet, bevor mit dem Booten fortgefahren wird. Drückt der Benutzer innerhalb dieser Zeit die Shift-Taste , so erscheint das Auswahlmenü.

GRUB_TIMEOUT hat keine Auswirkung, falls **GRUB_HIDDEN_TIMEOUT** gesetzt ist. Ausnahme: Hat diese Variable den Wert `-1`, so erscheint das Auswahlmenü immer.

GRUB_CMDLINE_LINUX der Inhalt dieser Variablen wird dem Linux-Kernel aller Konfigurationen (Normal und Recovery) als Argument mitgegeben.

GRUB_CMDLINE_LINUX_DEFAULT der Inhalt dieser Variablen wird dem Linux-Kernel der Normal-Modi mitgegeben.

2. Falls nicht ausreichend, ergänze Shell-Skripte unterhalb von `/etc/grub.d/`, welche die Konfigurationsdatei des GRUB 2 generieren. Die Dateien in diesem Verzeichnis haben folgende Aufgaben:

00_header generiert den Kopfteil von `/boot/grub/grub.cfg`.

05_debian_theme generiert Variablen zum Aussehen des Bootmenüs.

10_linux generiert die Booteinträge für das installierte Linux.

20_memtest86+ generiert einen Menüeintrag für das RAM-Test-Werkzeug **memtest86**.

30_os-prober generiert die Einträge für weiter installierte Betriebssysteme (Linux, Windows, OS X).

40_custom Hier kann der Administrator eigene Menüeinträge hinzufügen.



Beispiel:

```
#!/bin/sh
exec tail -n +3 $0
# This file provides an easy way to add custom menu entries.
# Simply type the menu entries [...] after this comment.
# Be careful not to change the 'exec tail' line above.
menuentry 'Mein Linux' {
    set root='(hd0,2)'
    linux    /boot/vmlinuz-3.0.0-17 root=/dev/sda2 ro
    initrd  /boot/initrd.img-3.0.0-17
}

menuentry 'Windows 8' {
    set root='(hd0,3)'
    chainloader (hd0,3)+1
```

```
}
```

3. Nun können wir testweise ansehen, wie die resultierende Konfigurationsdatei aussehen würde:

```
# grub-mkconfig
#
# DO NOT EDIT THIS FILE
#
# It is automatically generated by grub-mkconfig using templates
# from /etc/grub.d and settings from /etc/default/grub
...
```

4. Dann wird aus den Konfigurationsdateien der vorhergehenden Schritte die Hauptkonfigurationsdatei generiert:

```
# update-grub
Generating grub.cfg ...
Found linux image: /boot/vmlinuz-3.0.0-17-generic-pae
Found initrd image: /boot/initrd.img-3.0.0-17-generic-pae
...
Found memtest86+ image: /boot/memtest86+.bin
done
```

Unter Fedora 16 würde dieser Schritt mangels des **update-grub** so aussehen:

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

Ist dagegen GRUB 2 noch gar nicht im (Master-) Bootsektor der Platte/Partition installiert, so existiert dafür wie unter GRUB auch das Kommando **grub-install**. Unter RedHat-Derivaten heißt dieses Kommando **grub2-install**.

1.3.2 Passwortschutz in GRUB 2 einrichten

Will man verhindern, dass beliebige Nutzer die Kernel-Optionen beim Booten verwenden, so kann man dafür ein GRUB-seitiges Passwort konfigurieren. Dazu fügt man am Ende der Datei `/etc/grub.d/00_header` folgende Zeilen an:

```
cat <<EOF
# Erlaubt dem User superman Editieren und Kommandozeile
set superusers="superman"
# Passwort des Benutzers superman
password superman 1234
# Einrichten eines normalen Benutzers
password bill 5678
EOF
```

Hier wurde außerdem ein normaler Benutzer angelegt. Dieser darf zwar weder Bootinträge editieren, noch die Kommandozeile ausführen, jedoch kann man bestimmte Menüeinträge mit Passwort versehen. So können wir z.B. dafür sorgen, dass nur der Benutzer `bill` das Betriebssystem Windows booten darf:

```
menuentry -users bill 'Windows 8' {
    set root='(hd0,3)'
    chainloader (hd0,3)+1
}
```

Natürlich wird bei dem obigen Beispiel dem sicherheitsbewussten Administrator sofort aufstoßen, dass das Passwort hier in Klartext abgelegt wurde. Um hier Abhilfe zu schaffen, erzeugt man mit dem Befehl **grub-mkpasswd-pbkdf2** ein verschlüsseltes Passwort:



Verschlüsseltes Passwort für GRUB 2 einrichten

1. Generieren des Passwort-Hashes:

```
# grub-mkpasswd-pbkdf2
Enter password: Passwort eingeben
Reenter password: Passwort nochmals eingeben
Your PBKDF2 is grub.pbkdf2.sha512.10000.70470C47F420C45A2...
```

Unter RedHat-Derivaten lautet der Befehl **grub2-mkpasswd-pbkdf2**.

2. Den Eintrag in `00_header` ändert man dann wie folgt ab:

```
# Passwort des Benutzers superman
password_pbkdf2 superman grub.pbkdf2.sha512.10000.70470C47...
```

1.4 Runlevel und das System-V-Init-System

Der `init`-Prozess ist sozusagen der Vater aller Prozesse (LPI 1: 101.3). Auf den meisten UNIX-Systemen der Welt hat er die Prozess-ID 1 (nicht z.B. bei Compaq Tru64 UNIX im Cluster). Der `init`-Prozess ist der erste Prozess, der beim Hochfahren des Systems vom Kernel gestartet wird. Seine Hauptaufgabe ist die Erzeugung anderer Prozesse, die wiederum durch spezielle Skripte gesteuert werden. Dies wird auf modernen Linux-Systemen mittlerweile von einem modernen Init-System wie z.B. **systemd** oder **upstart** erledigt. Das herkömmliche System-V-Init-System, welches aber noch in vielen Installationen und manchen Distributionen anzutreffen ist, wird klassischerweise in der Datei `/etc/inittab` konfiguriert. Ⓛ

Im Rahmen dieses Kapitels wird die Struktur des System-V-Init-Systems (abgekürzt SysV-Init) behandelt, da es selbst auf modernen Distributionen mittels eines Rückwärtskompatibilitätsmechanismus von **systemd** und **upstart** bei vielen Softwarepaketen verwendet wird.

Das aus dem UNIX System V stammende Konzept des verteilten Startup-Mechanismus wurde auch in Linux übernommen. Dieses Konzept erlaubt eine flexible Konfiguration der Systemprozesse und Anwendungen. So wurde bei älteren UNIX-Systemen lediglich zwischen einem Single-User- und einem Multi-User-Modus unterschieden. Durch Definition verschiedener Runlevel hingegen kann man genau spezifizieren, ob zum Beispiel die grafische Benutzeroberfläche standardmäßig aktiviert ist oder nicht (auch schon zum Anmelden).

Theoretisch werden die Runlevel 0 bis 9 und `s` unterstützt. In der Praxis werden jedoch meist nur die Runlevel 1 bis 6 benutzt. In der Datei `/etc/inittab` werden die oben erwähnten Skripte und andere Aktionen für die jeweiligen Runlevel und auch der Default-Runlevel festgelegt.

Da diese Datei normalerweise nur beim Starten des Systems und beim Wechsel des Runlevels gelesen wird, muss man nach jeder Änderung, die sofort wirksam werden soll, den aktuellen Runlevel mit dem Befehl `init q` neu initialisieren. Ein Runlevel wird gestartet, indem man den gewünschten Level als Argument an das `init`-Kommando übergibt: `init runlevel` oder `telinit runlevel`, z.B. `init 5` oder `telinit 5` (LPI 1: 101.2) (LPI 1: 101.3) Ⓛ
Ⓛ

Die Runlevel haben folgende Bedeutung:

Runlevel	Bedeutung
0	Der Runlevel 0 dient zum geordneten Herunterfahren des Systems. Prozesse und Applikationen können sauber beendet werden, und die Filesysteme werden sauber verlassen (Wegschreiben des Buffer Cache).
s oder S	Single-User-Modus mit US-Tastaturbelegung
1	Single-User-Modus
2	Multi-User-Modus ohne Netzwerk
3	Multi-User-Modus mit Netzwerk
4	nicht benutzt (Standardmäßig wie Runlevel 3)
5	Multi-User-Modus mit Netzwerk und grafischem Login
6	Reboot

Mit dem Kommando **runlevel** kann man sich den vorherigen und den aktuellen Runlevel anzeigen lassen. **N** bedeutet dabei: kein vorheriger anderer Level.

```
# runlevel
```

```
N 2
```

Single-User-Modus Der Single-User-Modus wird nur benötigt, wenn man Systempflege betreiben will, z.B. Reorganisation der Filesysteme, oder eine gründliche Datensicherung durchführen möchte. Im Single-User-Modus kann kein anderer Benutzer außer *root* interaktiv arbeiten. Es stehen keine virtuellen Konsolen und kein Netzwerk zur Verfügung (LPI 1: 101.3).



Vor dem Hochfahren des Systems kann im Bootprompt (falls aktiviert) das Schlüsselwort *single* angeben, um nur in den Single-User-Modus zu gelangen:

```
GRUB boot: linux single
```

Bei der SuSE-Distribution ist dann nur das Root-Filesystem / im Read-Only-Modus und das Pseudofilesystem /*proc* eingehängt. Das Root-Passwort *muss* angegeben werden.

Bei Fedora/RedHat sind alle Filesysteme normal eingehängt, und man benötigt kein Kennwort.

Zum Herunterfahren in den Single-User-Modus verwendet man **init s**. In diesem Fall bleiben die lokalen Dateisysteme eingehängt; das Netzwerk wird gestoppt.

1.5 /etc/inittab

- Ⓛ Als Beispiel für die Datei /etc/inittab (LPI 1: 101.3) die Standardversion der SuSE-Distribution:

```
1 #
2 # inittab      Diese Datei beschreibt, wie INIT das System in einen
3 #              bestimmten Runlevel hochfährt
4
5 # Default Runlevel (der nach dem Booten zu startende Runlevel):
6 # 0 - halt (System herunterfahren, NICHT als Default Runlevel!)
7 # 1 - Single user mode (Ein-Benutzer-Modus)
8 # 2 - Multiuser, ohne NFS
9 # (The same as 3, if you do not have networking)
10 # 3 - Voller Multiuser-Modus
11 # 4 - unused
12 # 5 - X11-Grafisches Login
13 # 6 - reboot (System neu starten, NICHT als Default Runlevel!)
14 #
15 id:5:initdefault:
16
17 # Grundlegende Systeminitialisierung, wie Mounten der Dateisysteme,
18 # Dateisystem-Check,
19
20 si::sysinit:/etc/rc.d/rc.sysinit
21
22 10:0:wait:/etc/rc.d/rc 0
23 11:1:wait:/etc/rc.d/rc 1
24 12:2:wait:/etc/rc.d/rc 2
25 13:3:wait:/etc/rc.d/rc 3
26 14:4:wait:/etc/rc.d/rc 4
27 15:5:wait:/etc/rc.d/rc 5
28 16:6:wait:/etc/rc.d/rc 6
29
30 # Bei jedem Runlevel-Wechsel ausführen, schreibt
31 # u.s. die Festplattencaches auf die Platte
32 ud::once:/sbin/update
33
34 # Das Kommando, das beim Dreifingerfriff ausgeführt wird.
35 ca::ctrlaltdel:/sbin/shutdown -t3 -r now
36
37 # Das Kommando, welches durch "Alt-Cursor up" gestartet wird.
38 kb:2345:kbrequest:/bin/sync
39
40 # Wenn eine Unterbrechungsfreie Stromversorgung verwendet wird,
41 # dann wird das System rechtzeitig heruntergefahren, bevor deren
```

```
42 # Batterie ausgeht.
43 pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"
44
45 # Wenn das Stromnetz zurückgekehrt ist, bevor der Shutdown ausgeführt wurde,
46 # dann wird der Shutdown abgebrochen
47 pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"
48
49
50 # Startet den Login-Prompt auf den virtuellen Terminals
51 1:2345:respawn:/sbin/mingetty tty1
52 2:2345:respawn:/sbin/mingetty tty2
53 3:2345:respawn:/sbin/mingetty tty3
54 4:2345:respawn:/sbin/mingetty tty4
55 5:2345:respawn:/sbin/mingetty tty5
56 6:2345:respawn:/sbin/mingetty tty6
57
58 # Startet das grafische Login im Runlevel 5
59 x:5:respawn:/etc/X11/prefdm -nodaemon
```

Aufbau der Datei /etc/inittab Die Einträge in der Datei /etc/inittab haben folgenden Aufbau:

ID:run-level:action:program

Die Felder haben folgende Bedeutung:

ID Zeichenkette zur Identifizierung des Eintrags

run-level Enthält alle Runlevel, in denen *program* gestartet werden soll

action Schlüsselwort, mit dem festgelegt wird, wie *program* ausgeführt werden soll

program Skript oder Kommando, das ausgeführt werden soll

Einige der wichtigsten Schlüsselwörter für das Feld *action*:

initdefault Legt den Standard-Runlevel fest, in obigem Beispiel Runlevel 2

boot Der Befehl wird nur beim Hochfahren des Systems gestartet

bootwait Wie oben, jedoch wartet der **init**-Prozess, bis *program* beendet ist, bevor der nächste Eintrag verarbeitet wird.

Starten und Anhalten des Systems

once *program* wird bei Eintritt in den/die angegebenen Runlevel gestartet. Seine Beendigung wird nicht abgewartet. Wird *program* beendet, wird es nicht erneut gestartet.

wait Wie oben, jedoch wartet der **init**-Prozess, bis *program* beendet ist, bevor der nächste Eintrag verarbeitet wird.

respawn Bei einem Wechsel des Runlevel wird der Prozess gestartet, falls er noch nicht existiert. Sobald er beendet ist, wird er neu gestartet.


Wird in der Regel nur für Programme verwendet, die einen Login-Prompt zur Verfügung stellen und nach jedem Ausloggen eines Benutzers neu gestartet werden sollen (z.B. **getty** für die virtuellen Konsolen und **xdm** für den grafischen Login).

off Der Prozess wird beendet, wenn der Runlevel, z.B. mit **init q**, neu initialisiert wird.

powerfail Befehle, die ausgeführt werden sollen, falls die USV einen Stromausfall meldet.

ctrlaltdel steuert das Verhalten, wenn die Tastenkombination Strg-Alt-Entf gedrückt wird. Voreinstellung: kontrollierter Neustart des Systems.

kbrequest erlaubt die Belegung der Kombination Alt-↑ mit Programmen als Shortcut. Dies funktioniert jedoch nur auf den Textkonsolen, nicht im X-Window-System!

Hinweis: Bei modernen Distributionen mit **systemd** oder **upstart** existiert die Datei `/etc/inittab` nicht mehr. Sie wurde durch die Konfigurationsdateien des entsprechenden Init-Systems ersetzt. 

1.6 Verteilter Startup-Mechanismus

Wie man in dem vorangehenden Beispiel der `/etc/inittab` sehen kann, wird beim Wechsel des Runlevels im Wesentlichen das Kommando `/etc/init.d/rc runlevel` ausgeführt:

```
...
# Default runlevel. The runlevels used by RHS are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (same as 3, if you don't have network)
# 3 - Full multiuser mode
```

```
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
#
id:5:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

10:0:wait:/etc/rc.d/rc 0
11:1:wait:/etc/rc.d/rc 1
12:2:wait:/etc/rc.d/rc 2
13:3:wait:/etc/rc.d/rc 3
14:4:wait:/etc/rc.d/rc 4
15:5:wait:/etc/rc.d/rc 5
16:6:wait:/etc/rc.d/rc 6
...
```

Dahinter verbirgt sich ein einfacher, aber genialer Mechanismus:



- Im Verzeichnis `/etc/init.d/` (LPI 1: 101.3) befinden sich mehrere Skripte, die bestimmte Systemprozesse starten oder stoppen können, je nachdem, ob sie mit dem Argument `start`, `stop` oder `restart` aufgerufen werden.
- Außerdem gibt es Verzeichnisse `/etc/rc.d/rcn.d` für jeden Runlevel, also z.B. `rc3.d` für Runlevel 3.
- In diesen Unterverzeichnissen liegen *symbolische Links* auf die für diesen Runlevel benötigten Skripte. Die Namen dieser Links beginnen mit `K` oder `S` und einer Nummer zur Sortierung (Festlegung der *Reihenfolge*). Diese Links werden in alphanumerischer Reihenfolge nach der ASCII-Tabelle abgearbeitet.
- Der „Master Resource Control Script“ `/etc/rc.d/rc` durchläuft nun in einer Schleife das Verzeichnis des gewünschten Runlevels, wobei zuerst alle Skripte, deren Name mit `K` beginnt, mit dem Argument `stop` ausgeführt werden („kill“), und dann alle Skripte, deren Name mit `S` beginnt, mit dem Argument `start` ausgeführt werden.
- *Nur bei alten SuSE-Versionen bis einschl. 8.0:* Wichtige Parameter und Definitionen für das Skript `/etc/init.d/rc` befinden sich in Konfigurationsdateien im Verzeichnis `/etc/rc.config.d/`.