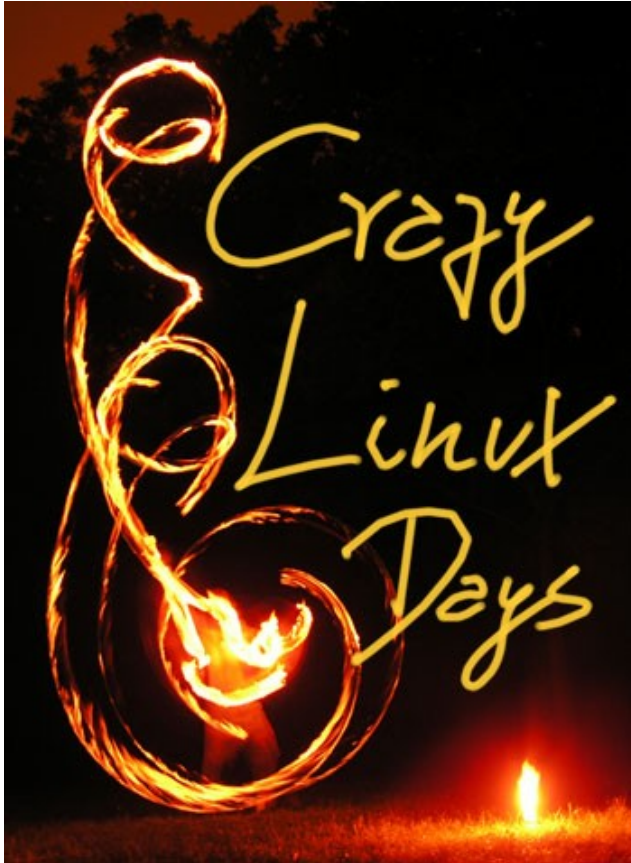


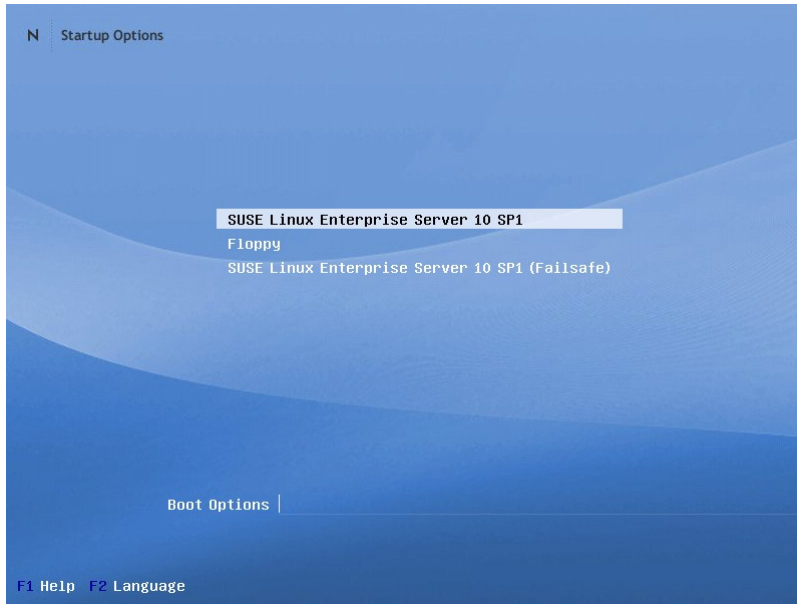


- Troubleshooting

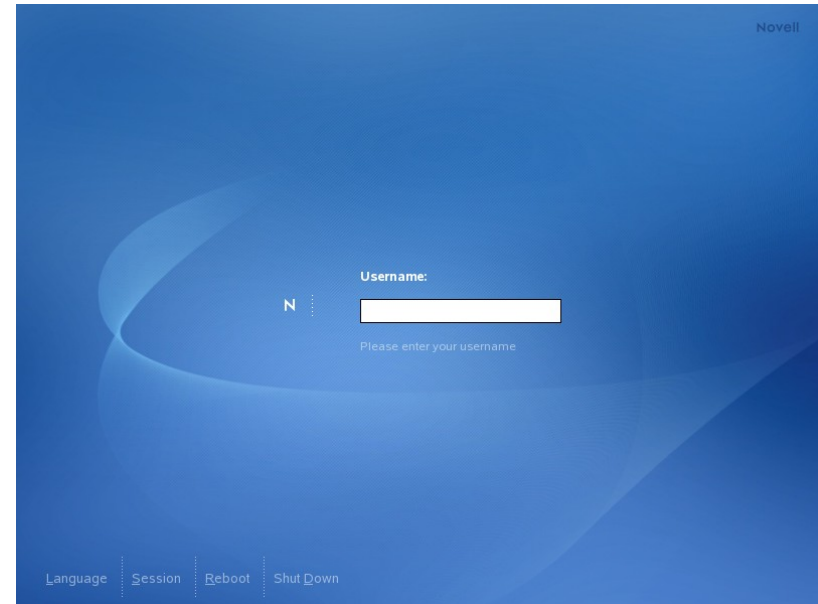
Linux Bootprozess

1. Bootprozess im Überblick
2. MBR & Partitionstabelle
3. Grub Bootloader
4. Init Alternativen
5. Kernel & Kernelmodule

Bootloader Menü



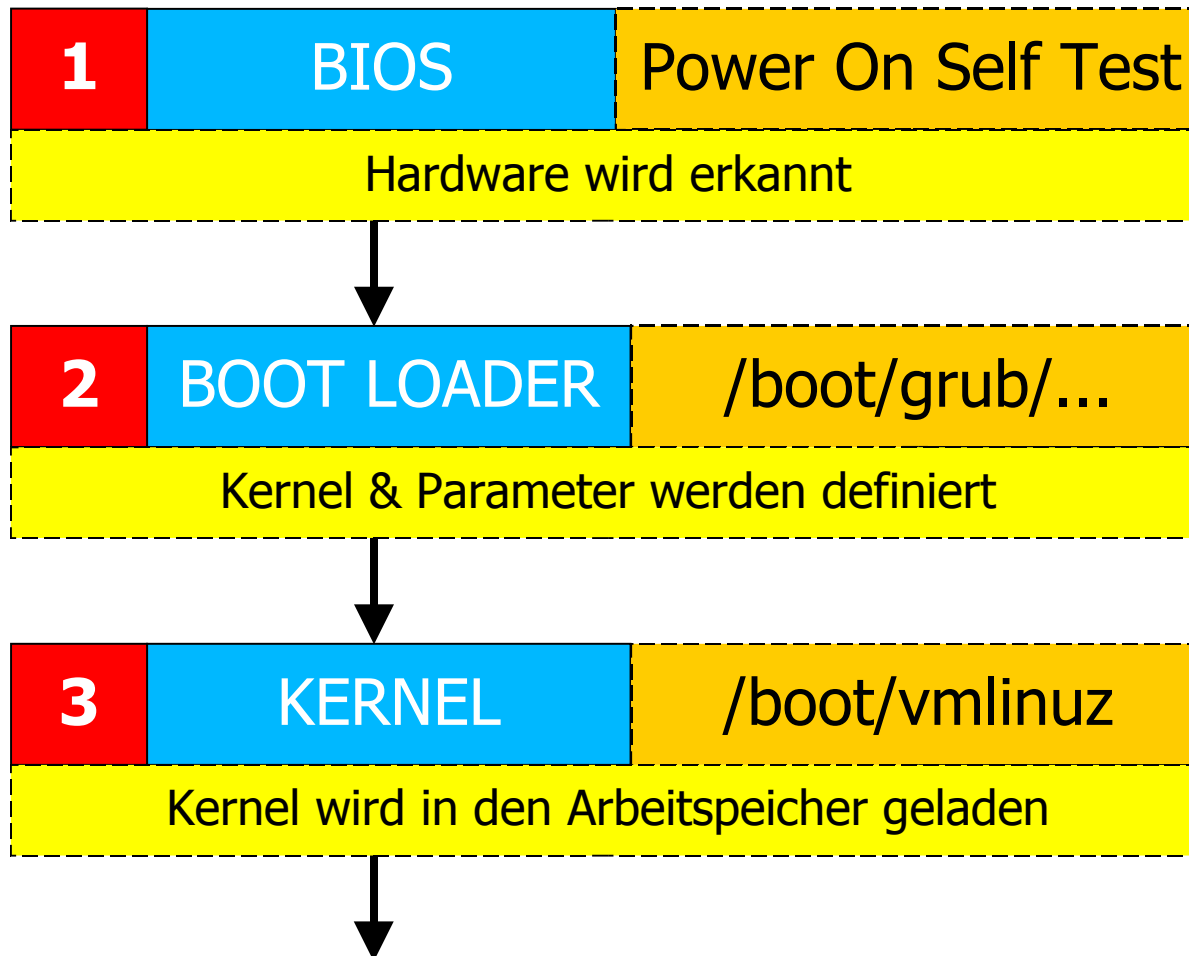
Loginmaske

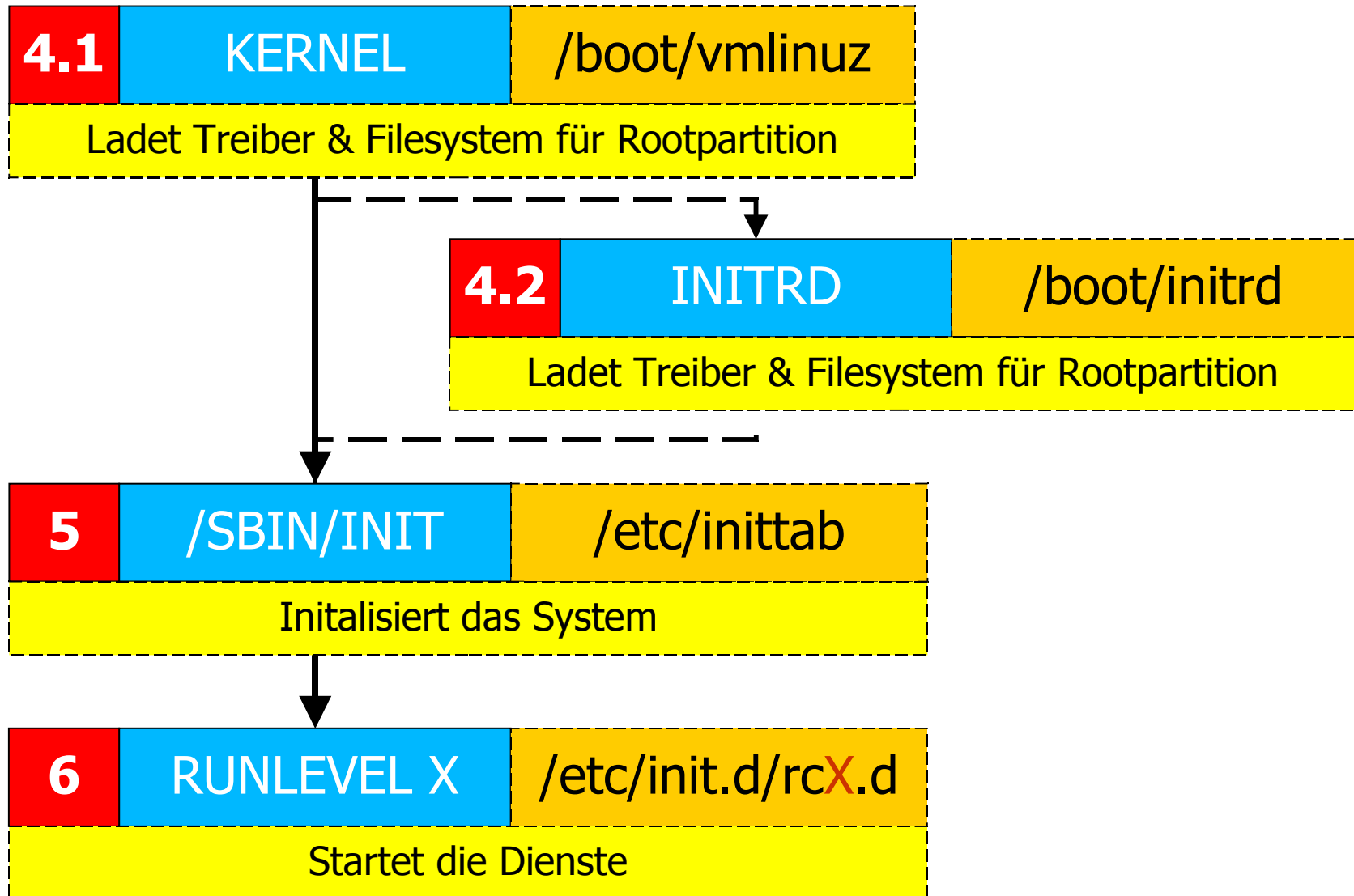


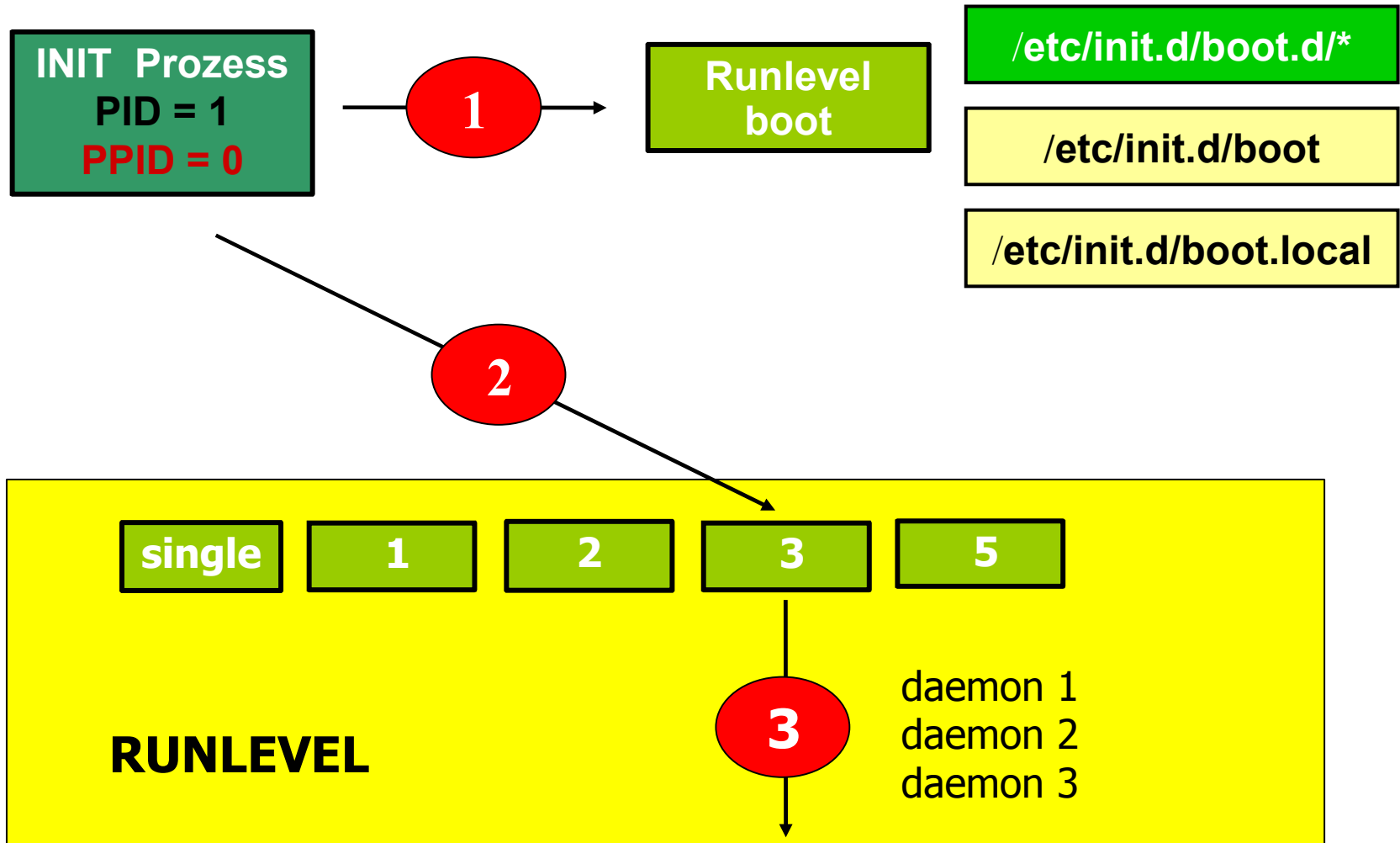
28 sek. - 55 sek.

- Wie sieht der Bootprozess im Überblick aus?

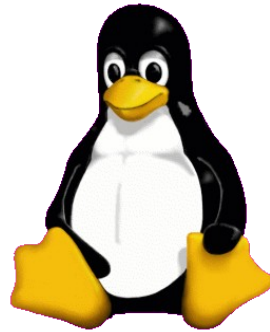








- Der POST lässt sich in einzelne Schritte einteilen:
 - Überprüfung der Funktionsfähigkeit der CPU
 - Überprüfung der CPU-nahen Bausteine
 - Überprüfung des CMOS-RAM (Prüfsummen-Bildung)
 - Überprüfung des CPU-nahen Cache-Speichers
 - Überprüfung der ersten 64 Kilobyte des Arbeitsspeichers
 - Überprüfung des Grafik-Speichers und der Grafik-Ausgabe-Hardware
 - Auswahl des Boot Gerätes

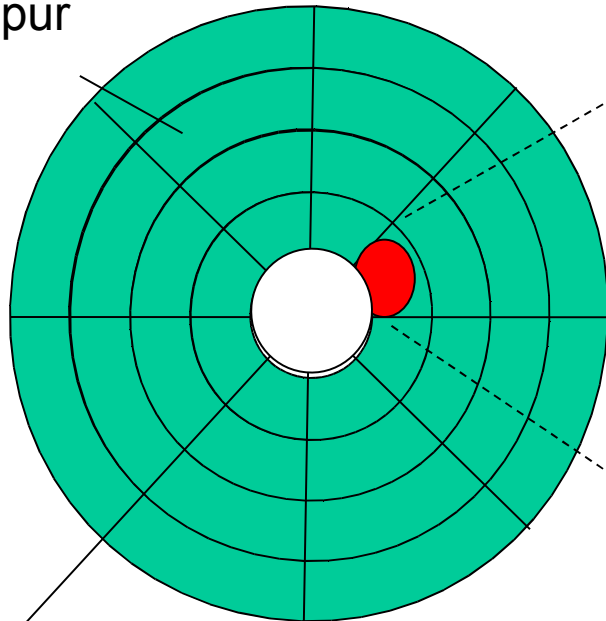


Master Boot Record

- Details zum MBR

- ist der erste Sektor auf der Festplatte
- Disk 0, Head 0, Sector 1
- 512 Byte gross

Spur



Sektor

MASTER BOOT CODE

GRUB Bootloader STAGE 1

PARTITION TABLE

Partition Nr.	Start	Ende	...
1	1	190	
2	191	227	
3	228	405	
4	406	903	

4 Partitionen

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1B0																
1C0			83													
1D0			83													
1E0			8E													
1F0			82												55	AA



1. Partition 0x1BE - 0x1CD

2. Partition 0x1CE - 0x1DD



3. Partition 0x1DE - 0x1ED

4. Partition 0x1EE - 0x1FD

- Backup des Master Boot Records erstellen

```
dd if=/dev/sda of=/tmp/mbr bs=512 count=1
```

- Sichert den Bootcode (1-446 Byte)
- Sichert die prim. & erweiterte Partition (447-512)
- Sichert KEINE logischen Laufwerke!!!

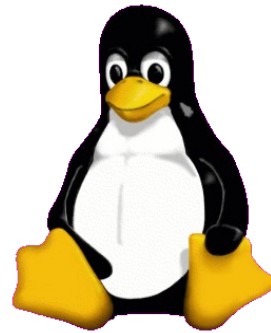
- Sichern aller Partitionsdaten

`sfdisk -d /dev/hda > hda.out`

- Sichert die prim. & erweiterte Partition
- Sichert alle logischen Laufwerke
- Sichert keinen Bootloader Code

- Wiederherstellen des MBR & Partitionsdaten

`sfdisk /dev/hda < hda.out`



GRUB Bootloader

- Aufgabe eines Bootloaders
 - Verschiedenste OS zu starten
 - Kernel mit Parametern starten
 - Kernelfunktionen
 - Hardware Einstellungen
 - Runlevel
- Bootloader Arten
 - LILO
 - GRUB



- Welcher Boot Loader wird verwendet?

/etc/sysconfig/bootloader

LOADER_TYPE="grub"

- GRUB Konfigurationsdateien

/boot/grub/menu.lst

/etc/grub



- Aufbau von GRUB

- GRUB ist modular aufgebaut da der MBR in der Größe sehr beschränkt ist

stage1 ...primäre Bootloader

stage1.5 ...Brücke zwischen stage 1 & 2

stage2 ...sekundäre Bootloader

● GRUB Stage 1

- liest nun Stage 1.5 oder 2 aus einer beliebigen Partition aus, wobei die genaue Position in einer Map-Datei steht. (kein Dateisystemtreiber)
- kann existieren im **mbr, cdrom, floppy, usb**
- Größe beschränkt auf max 446 bytes

Eine Kopie von Stage 1 liegt unter **/boot/grub/stage1**

damit können auch andere Bootgeräte erzeugt werden

- `/boot/grub/stage2`

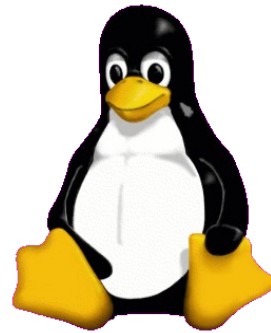
- enthält nun die Dateisystemtreiber sowie die eigentliche Shell, den Code für das Menü und den Loader für den Kernel.
- Nach dem Laden dieser Datei wird die Konfigurationsdatei `/boot/grub/menu.lst` eingelesen
- Der Benutzer kann nun in der Konsole das zu bootende Betriebssystem aus einem Menü auswählen oder Befehle in den GRUB-Shell eingeben.
- Dieser Teil ist der eigentliche Bootloader, von welchem aus der Kernel oder ein Bootblock innerhalb einer Partition geladen wird.

● Stage 1.5

- Diese Lösung hatte immer noch den Nachteil, dass der Bootloader nach Verschieben oder Ändern von Stage 2 nicht mehr bootfähig war.
- Deswegen wurde zwischen Stage 1 und 2 eine Zwischenstufe, Stage 1.5, eingeführt welcher zwischen dem MBR und den ersten Datenblöcken der ersten Partition liegt
- Stage 1.5 bietet genau eine Fähigkeit nämlich das Dateisystem des /boot Verzeichnisses zu lesen.

e2fs_stage1_5	iso9660_stage1_5	reiserfs_stage1_5
fat_stage1_5	jfs_stage1_5	ufs2_stage1_5
xfs_stage1_5	minix_stage1_5	vstafs_stage1_5

- Zur Zeit gibt es Stage 1.5 für die Dateisysteme
 - FAT, Minix, Ext2, JFS, ReiserFS sowie XFS.
 - Unterstützung für Reiser4 gibt es in einem Extra-Patch.



Konfigurationsdatei

- Grub Konfigurationsdatei

- /boot/grub/menu.lst ...SUSE
- /boot/grub/grub.conf ...RedHat

```
default 0
```

```
timeout 8
```

```
title linux
```

```
kernel (hd0,0) /boot/vmlinuz root=/dev/hda1
```

```
initrd (hd0,0) /boot/initrd
```

Wo findet Grub den Kernel

kernel (hd0,0) /boot/vmlinuz root=/dev/hda1

Festplatte, Partition

Kerneldatei

Root-Dateisystem

Wo findet Grub das passende Initial Ramdisk

initrd (hd0,0) /boot/initrd

Festplatte, Partition

Initrd-datei

```
kernel (hd0,0) /boot/vmlinuz root=/dev/hda1
```

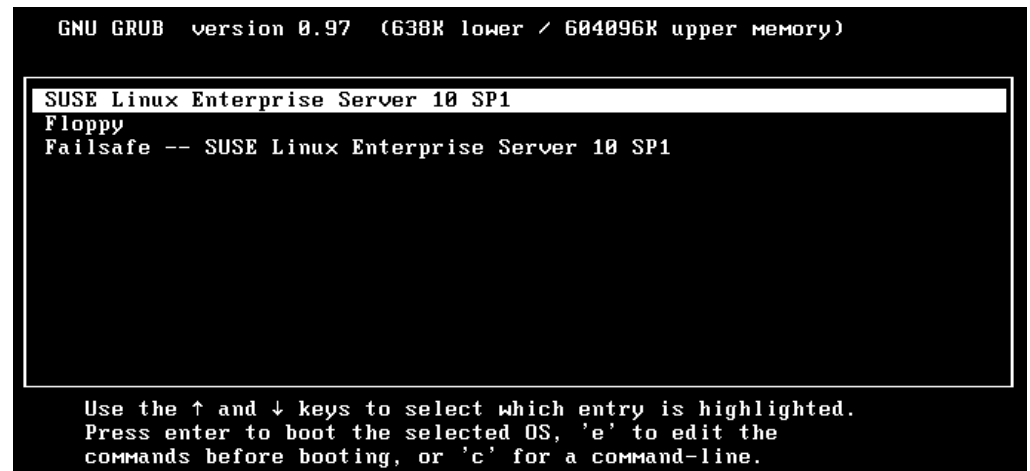
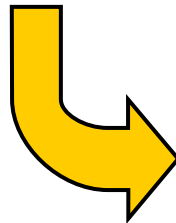
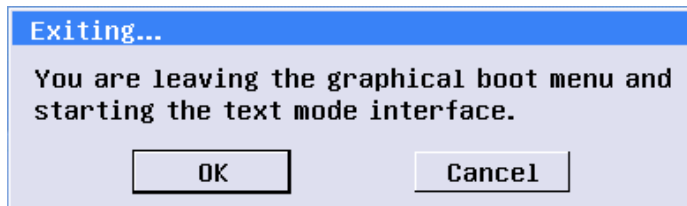
- GRUB unterscheidet nicht zwischen IDE und SCSI
- die Festplatten werden mit (hd...) bezeichnet
- GRUB zählt Partitionen ab *Null*, während fast alle anderen Betriebssysteme von Eins zählen

hd0,0 ...erste HD, erste Partition

hd1,3 ...zweite HD, vierte Partition

- Diskettenlaufwerke werden mit (fd...) bezeichnet.
- BIOS Bootreihenfolge verändert die Nummerierung!

- Wechseln zur Grub Text Konsole
 - GRUB Grafik Modus mit **ESC** verlassen



- Temporäres verändern der Bootparameter
 - richtige OS Zeile auswählen und **E** drücken
 - Zeile auswählen und **E** drücken zum editieren
 - **ESC** zurück zum Bootmenü

```
GNU GRUB  version 0.97  (638K lower / 604096K upper memory)

root (hd0,1)
kernel /vmlinuz-2.6.16.46-0.12-default root=/dev/sda5 vga=0x332  re→
initrd /initrd-2.6.16.46-0.12-default
```

- Wechseln zur Grub Shell
 - C drücken
 - help

```
GNU GRUB  version 0.97  (638K lower / 604096K upper memory)

[ Minimal BASH-like line editing is supported.  For the first word, TAB
  lists possible command completions.  Anywhere else TAB lists the possible
  completions of a device/filename.  ESC at any time exits. ]

grub> _
```

- Shell kann mit **ESC** wieder verlassen werden

Inhalt von /etc/fstab

peter@asterix:~> cat /etc/fstab

/dev/sda1	swap	swap	defaults	0 0
/dev/sda2	/boot	ext2	acl,user_xattr	1 2
/dev/sda3	/	ext3	acl,user_xattr	1 1
/dev/sda4	/home	reiserfs	acl,user_xattr	1 2

- Auf welcher Partition liegt der Kernel?
- Auf welcher Partition liegt das Root-Dateisystem?

GRUB Shell

grub> find /boot/vmlinuz ...entspricht /boot-Partition (sda2)
(hd0,1)

grub> find /etc/fstab ...entspricht /-Partition (sda3)
(hd0,2)

Inhalt von /etc/fstab

```
peter@asterix:~> cat /etc/fstab
```

/dev/sda1	swap	swap	defaults	0 0
/dev/sda2	/boot	ext2	acl,user_xattr	1 2
/dev/sda3	/	ext3	acl,user_xattr	1 1
/dev/sda4	/home	reiserfs	acl,user_xattr	1 2

Grub Konfigurationsdatei

```
asterix:~# cat /boot/grub/menu.lst
```

```
...
```

```
title Kernel-2.6.18.8-0.1-default
```

```
    root (hd0,1)
```

```
    kernel /vmlinuz-2.6.18.8-0.1-default root=/dev/sda3 vga=0x317
```

```
    initrd /initrd-2.6.18.8-0.1-default
```

- Partitionen finden

- `find /boot/vmlinuz` ...wo liegt der Kernel
- `find /etc/fstab` ...wo ist die Root Partition

- Systemeinstellungen definieren

- `root (hd0,1)` ...Kernel Partition definieren
- `kernel /boot/vmlinuz root=/dev/sda3`
- `initrd /boot/initrd`

- System starten

- `boot`



• Starten des Kernel mit anderen Parametern

- Parameter werden durch den Bootloader übergeben
- Manche Parameter werden an init weitergereicht
- Erklärung der wichtigsten Kernelparameter unter:

</usr/src/linux/Documentation/kernel-parameters.txt>

#Mit welchen Parametern wurde der Kernel gestartet?

peter@tux> less /proc/cmdline

root=/dev/hda3 vga=0x317 selinux=0 resume=/dev/hda1 splash=silent

- 3 ...in Runlevel 3 starten
- confirm ...RC-Skripte einzeln bestätigen (Y/N)
- nocoldplug=1 ...coldplug deaktivieren
- nohotplug=1 ...hotplug deaktivieren
- vnc=1 ...VNC Server aktivieren
- acpi=off ...ACPI deaktivieren
- apm=off ...APM deaktivieren
- mem=512M ...nur 512MByte RAM benutzen

- Schrittweises Starten des Systems

- /etc/sysconfig/boot

PROMPT_FOR_CONFIRM="no"

CONFIRM_PROMPT_TIMEOUT="5"

```
Adding 514040k swap on /dev/sda1. Priority:-1 extents:1 across:514040k done
Start bootup script boot.proc, (Y)es/(N)o/(C)ontinue? [y]
Start bootup script boot.udev, (Y)es/(N)o/(C)ontinue? [y]
```

- Einmaliges Schrittweises Starten

- Grub Sektion mit Parameter **confirm** starten

```
Boot Options confirm|
```

- Dauerhafte Änderung
 - Eintrag in /etc/inittab anpassen

```
#Auszug aus /etc/inittab  
peter@tux # less /etc/inittab  
# The default runlevel is defined here  
id:5:initdefault:
```

- Temporäre Änderung in GRUB Menü
 - Bootoption: **single, 1, 2, 3** oder **5** angeben
 - Sektion anpassen: `root=/dev/hda4 vga=791` **1**

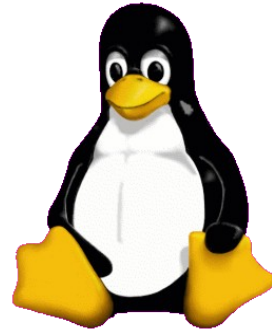


Boot Options 3 |

- Definieren der Bildschirmauflösung

- für eine Auflösung von 1024x768 einfach den bootparameter `vga=791` oder `vga=0x317` nehmen. Nicht alle bootparameter funktionieren überall. Bis 1024x768 sollte jedoch immer gehen.

Codetabelle für VGA							
col/res	640x400	640x480	800x600	1024x768	1152x864	1280x1024	1600x1200
2 (4bit)	x	x	(770)	x	x	x	x
256 (8bit)	0x300 (768)	0x301 (769)	0x303 (771)	0x305 (773)	0x161 (353)	0x307 (775)	0x31C (796)
32k (15bit)	x	0x310 (784)	0x313 (787)	0x316 (790)	0x162 (354)	0x319 (793)	0x31D (797)
64k (16bit)	x	0x311 (785)	0x314 (788)	0x317 (791)	0x163 (355)	0x31A (794)	0x31E (798)
16M (24bit)	x	0x312 (786)	0x315 (789)	0x318 (792)	x	0x31B (795)	0x31F (799)
16M (32bit)	x	x	x	x	0x164 (356)	x	x



INIT-Alternativen

• Aufgabe des INIT Prozesses

- ist der erste Prozess der nach dem laden des Kernel gestartet wird und initialisiert das System
- hat immer die Prozess ID 1 und kann nicht mit kill beendet werden
- INIT ist der letzte Prozess der beim herunterfahren beendet wird
- Es gibt alternativen zu **/sbin/init**

INIT Prozess
PID = 1
PPID = 0

● Aufgabe von Bootchart

- Erstellt ein grafisches Poster welches einen Überblick über den Bootprozess zeigt
- Wichtige Grundlage um den Systemstart zu verbessern



Boot chart for paul (Wed Dec 1 08:23:07 CST 2004)

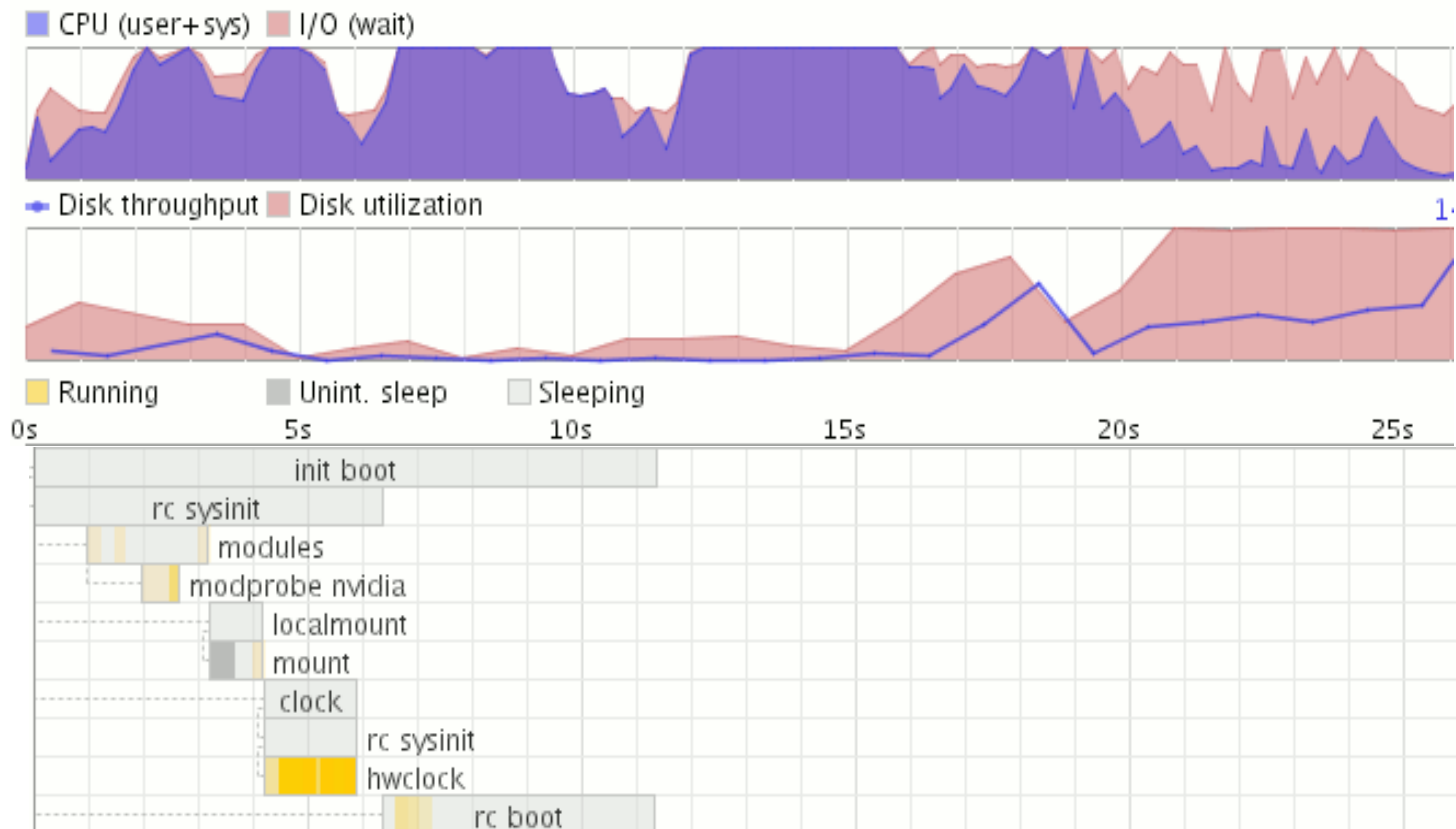
uname: Linux 2.6.9-gentoo-r1 #2 SMP Mon Nov 8 18:47:34 CST 2004 i686 GNU/Linux

release: Gentoo Base System version 1.6.6

CPU: Pentium III (Coppermine) (2)

kernel options: cmdline: root=/dev/ram0 init=/linuxrc real_root=/dev/sdb5 video=vesa:ywrap,mtrr vga=0x317

boot time: 0:36





- Notwendige Schritte

- RPM Paket von www.bootchart.org installieren
- Bootloader Konfiguration erweitern
- Ergebnis wird gespeichert als [/var/log/bootchart.tgz](#)

```
#/boot/grub/menu.lst
```

```
title linux
```

```
root (hd0,0)
```

```
kernel /boot/vmlinuz root=/dev/hda1 init=/sbin/bootchartd
```

```
initrd /boot/initrd
```

- Vorteile von /bin/bash als init-Prozess
 - kann zur Fehlerbehebung verwendet werden
 - Rücksetzen des Root User Kennwortes
 - Beheben von Konfigurationsproblem
 - Einfach im GRUB Menü als Startparameter `init=/bin/bash` definieren

```
Boot Options init=/bin/bash |
```

- Im GRUB Menü Startparameter definieren

`init=/bin/bash`

- Neues Kennwort definieren

`mount / -o remount,rw ...wenn es ro ist`

`passwd root`

`sync`

- Alternativ starten des Systems

`exec /sbin/init`



/lib ...Libraries

- Beinhaltet:

- Systemweite Libraries für /bin & /sbin
- Kernelmodule (/lib/modules)

#Anzeigen der verwendeten Libraries

```
peter@tux~> ldd /bin/cp  
linux-gate.so.1 => (0xffffe000)  
libacl.so.1 => /lib/libacl.so.1  
libattr.so.1 => /lib/libattr.so.1  
libselinux.so.1 => /lib/libselinux.so.1  
libc.so.6 => /lib/tls/libc.so.6
```



1.) Library Abhängigkeiten testen

`ldd /bin/ping`

2.) Umbenennen einer Library

`mv /lib/libc.so.6 /lib/libc.so.6.old`

3.) Library Abhängigkeiten erneut testen

`ldd /bin/ping`



- Stand-alone Shell (sash)

- Es kann vorkommen das wichtige Bibliotheken fehlerhaft werden z.B. `/lib/libc.so.6`
- Ohne diese Bibliothek funktionieren nicht einmal mehr Befehle wie `cd`, `mv` oder `cp`
- sash ist eine Shell die im Notfall verwendet werden kann um sein System wieder herzustellen

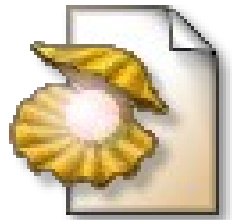
```
Boot Options init=/bin/sash|
```

- Funktionalitäten von sash

- ls, cd, cp, mv, mount, tar, gzip, gunzip, chown, chmod
- außer bei cd werden alle Befehle mit einem **Bindestrich** davor aufgerufen um sie von den normalen Befehle zu unterscheiden

-mount -cp -chmod

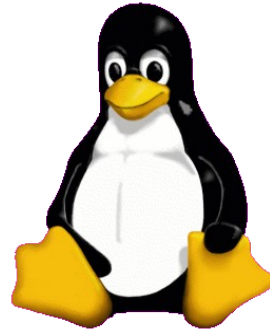
- Bootparameter: `init=/bin/sash`



- Wiederherstellen des Systems

- Bootparameter: `init=/bin/sash`
- `-cp /lib/libc.so.6.old /lib/libc.so.6`
- `sync`

`sync` schreibt den Cache sofort auf die HD damit beim abschalten des Systems keine Daten verloren gehen



Kernel & Kernelmodule

KERNEL

MODUL

MODUL

MODUL

MODUL

- **Modular Kernel (Version 2.x)**

- Kernel: /boot/vmlinuz
- Module: /lib/modules/*version*/kernel

- **Start Problem**

- Der Kernel ist modular aufgebaut
- Die Treiber für das Dateisystem werden normalerweise als Kernelmodul nachgeladen
- Damit der Kernel jedoch das Modul nachladen kann benötigt er bereits den Dateisystemtreiber für das Root Dateisystem (Henne-Ei Problem)

- **Lösung: *initrd***

● /boot/initrd

- beinhaltet alle für den Systemstart notwendigen Kernelmodule (Dateisystem-, Gerätetreiber,...)
- muss immer zu der aktuellen Kernelversion passen
- /boot/initrd ist ein Link zu `initrd-kernelversion`
- kann mit `mkinitrd` neu erzeugt werden
- Wenn die Root Partition gemountet ist wird `initrd`

```
# Inhalt einer entpackten initrd
asterix:/tmp/initramfs # ls
bin bootsplash dev etc init lib proc root sbin sys tmp var
```

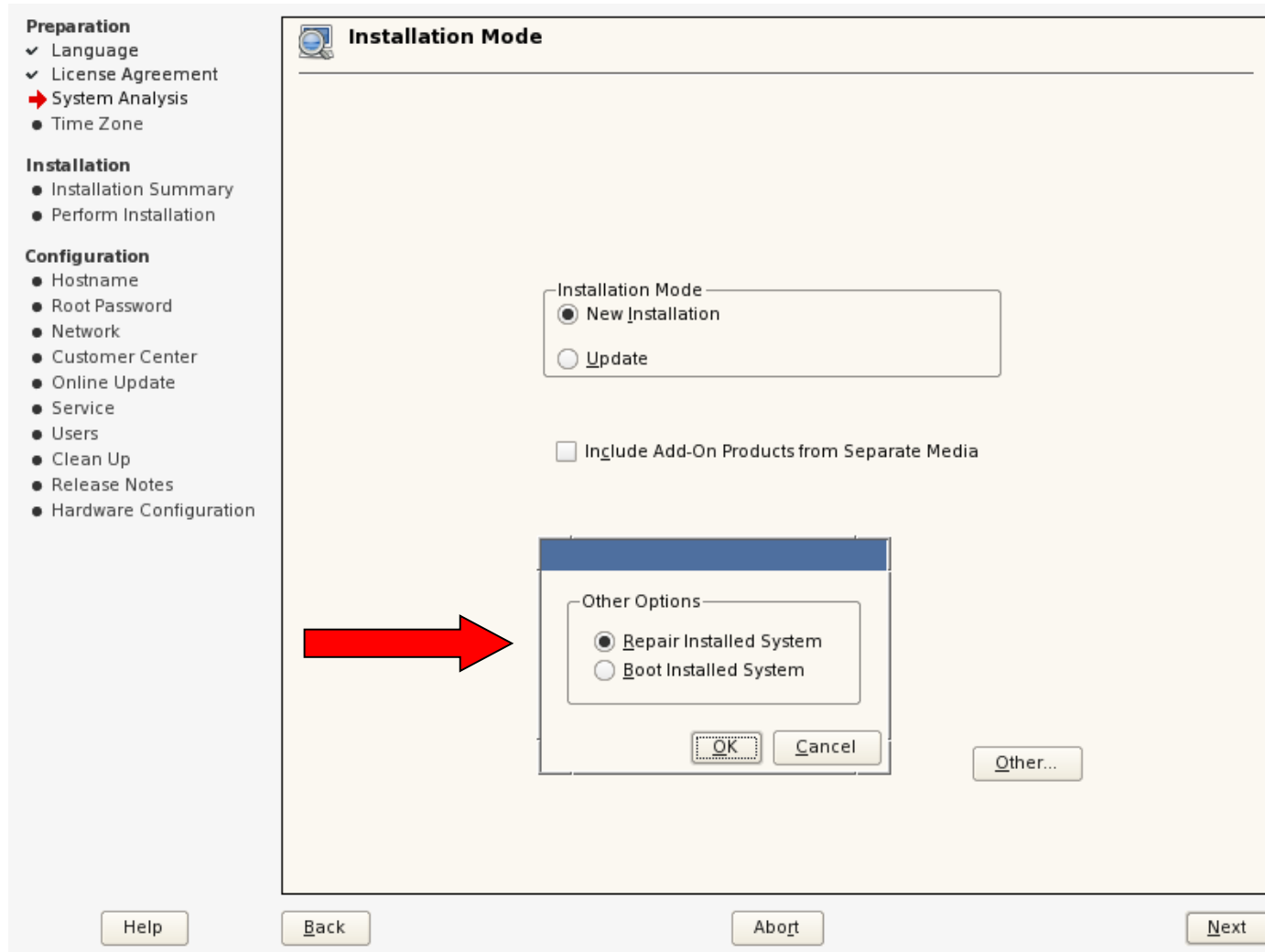
- Verändern der Module in initrd
 - In `/etc/sysconfig/kernel` Module hinzufügen
`INITRD_MODULES="xxx yyy zzz"`
 - `mkinitrd` neu ausführen

```
# Erweitern von initrd
peter@asterix:~# vi /etc/sysconfig/kernel
# This variable contains the list of modules to be added to the initial
# ramdisk by calling the script "mkinitrd"
# (like drivers for scsi-controllers, for lvm or reiserfs)
#
INITRD_MODULES="processor thermal ata_piix fan jbd ext3 edd"
```



Disaster Szenarien

- Reparatur mit der Installations DVD
 - Booten von der Installations DVD
 - Sprache auswählen
 - Lizenz akzeptieren
 - Am Schirm "Installationsmodus" den Punkt "Andere" auswählen
 - Reparatur des installierten Systems auswählen



- Wenn versucht wird das Root Dateisystem zu mounten kommt eine Kernel Panic Fehlermeldung

```
----- FEHLERMELDUNG -----  
Waiting for device /dev/sda1 to appear: . ok  
rootfs: major=8 minor=1 devn=2049  
rootfs: /sys/block/sda/sda1 major=8 minor=1 devn=2049  
mount: unknown filesystem type 'swap'  
umount: /dev: device is busy  
Kernel panic - not syncing: Attempted to kill init!  
Kernel panic: VFS: Unable to mount root fs on sda1  
----- FEHLERMELDUNG -----
```

- Mögliche Ursache

- Corrupt or misconfigured /boot/grub/menu.lst.

- Lösungsvorschlag

1. Boot Installed System*.
2. Login as root.
3. Edit /boot/grub/menu.lst.
4. Modify the kernel parameter root= to point to the correct root partition. fdisk -l should provide sufficient guidance to determine the root filesystem.

- Das System startet direkt in den Grub prompt
grub>

```
----- FEHLERMELDUNG -----  
KEINE  
----- FEHLERMELDUNG -----
```

- Mögliche Ursache

Falsch konfigurierte `/boot/grub/menu.lst`

- Lösungsvorschlag

1. Boot Installed System* -or- if sufficiently familiar with GRUB, manually boot the system.
2. Login as root. Verify the existence of `/boot/grub/menu.lst`. If existing, but misnamed, rename it. If corrupt, delete it and Repair Installed System** (just the Boot Loader Configuration check should be sufficient). If missing, Repair Installed System** (just the Boot Loader Configuration check should be sufficient).

- The system boots, but only to a GRUB screen, then hangs.

```
----- FEHLERMELDUNG -----  
GRUB Hard Disk Error  
----- FEHLERMELDUNG -----
```

- Mögliche Ursache

Da GRUB nicht komplett startet liegt das Problem bei
`/boot/grub/stage1`

- Lösungsvorschlag

1. Boot Installed System*.
2. Login as root.
3. Copy `/usr/lib/grub/stage1` to `/boot/grub/stage1`.
4. Reinstall GRUB: `grub-install /dev/sda`).
5. Reboot.

- Problembeschreibung

Egal welcher Kernel beim booten ausgewählt wird (failsafe or default), das System bricht mit einer Kernel Panic ab.

```
----- FEHLERMELDUNG -----  
RAMDISK: Couldn't find a valid RAM disk image starting at 0.  
VFS: Cannot open a root device "sda2" or unknown-block(0,0)  
Please append a correct "root=" boot option  
Kernel panic - not syncing: VFS:  
Cannot open a root device "sda2" or unknown-block(0,0)  
----- FEHLERMELDUNG -----
```

- Mögliche Ursache

Eine defekte oder fehlende `initrd`

- Lösungsvorschlag

1. Boot Installed System*.

2. Login as root.

3. Verify that the `/` (root) and `/boot` (if used) filesystems are mounted.

4. Run `mkinitrd`.

5. Reboot.

- Wenn das System bootet erscheint eine Fehlermeldung und das System friert ein
- Manchmal erscheint auch ein scharzer Schirm und das System rebootet automatisch

```
----- FEHLERMELDUNG -----  
No setup signature found ...  
initrd /boot/initrd  
[Linux-initrd @ 0x1fc38000, 0x2a7ab8 bytes]  
----- FEHLERMELDUNG -----
```

- Mögliche Ursache

Defekter Kernel im Verzeichnis /boot

- Lösungsvorschlag

1. Boot Installed System*.

2. Login as root.

3. Install a valid kernel rpm.

```
rpm -Uvh --force kernel-$type-$revision-$arch.rpm
```

4. Reboot.

- Das System startet nicht oder es startet und einige Module werden nicht geladen und manche Geräte werden nicht erkannt

```
----- FEHLERMELDUNG -----  
FATAL: Error inserting $modulename ($modulepath):  
Unknown symbol in module, or unknown parameter (see dmesg).  
$modulename: Unknown symbol $symbolname  
----- FEHLERMELDUNG -----
```

- Mögliche Ursache

Module wurden aktualisiert jedoch wurde die Datei
`/lib/modules/$kernelversion/modules.dep`
nicht aktualisiert bevor die initrd erstellt wurde

- Lösungsvorschlag

1. Boot Installed System*.
2. Run depmod.
3. Run mkinitrd.

it-iversity

Wolfgang Ziegler und Partner Ges.m.b.H.

Schottenfeldgasse 69

A – 1070 Vienna

Phone +43 (1) 5228222

servicecenter@it-iversity.com

© 2007 Wolfgang Ziegler und Partner GesmbH. All rights reserved.

This document / presentation is for informational purposes only.

WOLFGANG ZIEGLER UND PARTNER MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT and/or SUMMARY.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.