

# Initramfs - The initial RAM filesystem

## Am Beispiel von Debian

Dirk Geschke



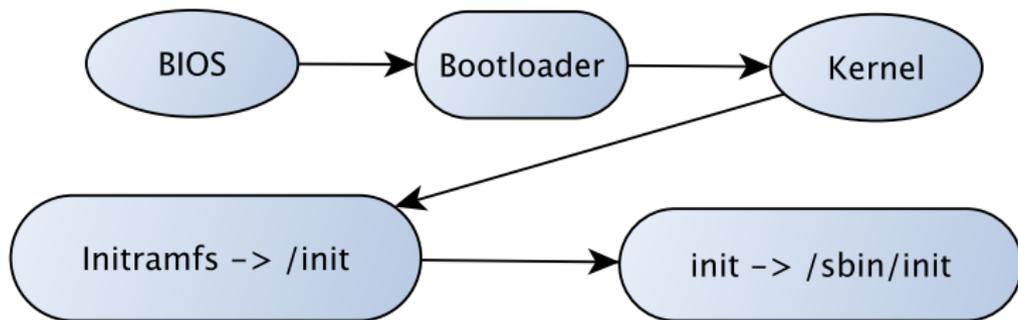
Linux User Group Erding

23. Mai 2012

# Gliederung

- 1 Einleitung
- 2 Unterschied zu initrd
- 3 Ablauf im Detail
- 4 Erstellen der initrd-Datei

# Überblick



# Allgemeines

- **initramfs** ist der Nachfolger von **initrd** ab Kernel 2.6
- **komprimiertes** Archiv mit benötigten Dateien für den Systemstart
- Wird beim **Booten** gemountet (`ramfs`) und `init` wird ausgeführt.
- Kann Module **laden**, Hardware **initialisieren** oder **vorbereiten** (per `udev`), Dateisysteme **entschlüsseln** oder auch das Root-Verzeichnis per **NFS** mounten.
- Startet dann das `init` im neuen **Root**-Verzeichnis

## Vergleich zu initrd

- **initrd** war separates Image (ext2-formatiert), **initramfs** wird ins Kernelimage gelinkt.
- initrd hatte **feste** Größe, initramfs nicht.
- initrd war gezipptes **Dateisystem**, der Treiber (ext2) musste im Kernel vorhanden sein.
- initrd musste in den **Kernel** zurückkehren, initramfs nicht

## Rund um Initramfs

- **cpio**: Standardformat, einfach und sauber
- gewöhnlich mit **gzip** gepackt
- liegt meistens in `/boot/`
- Inhalt ansehen:

```
gzip -dc initrd.img-2.6.32-5-amd64 | cpio -ivt
```

- Einstiegspunkt: `/init`, gewöhnlich ein **Shellskript**, kann aber auch Binärprogramm sein.

# Abarbeitung von `init`

- Erzeugen von Verzeichnissen: `/dev`, `/root`, `/sys`,  
`/proc`, `/tmp`, `/var/lock`
- mounten von `/sys` und `/proc`
- Erzeugen von `/dev/console` und `/dev/null`
- Ermitteln der Architektur: `/conf/arch.conf`
- Setzen von **Variablen**
- Auswerten von `/conf/initramfs.conf` → Module,  
keyboard, busybox, boot-Device, Netzwerkkarte

- Laden von Shellfunktionen: `/scripts/functions`
- Loggingfunktionen, `panic` (reboot mit Nachricht oder starten einer Shell), `run_scripts`, etc.
- Auswerten von `/conf/conf.d/` → z.B. Metadevices, Hostname, resume
- Auswerten der Bootzeile → `/proc/cmdline`

## Informationen aus Boot-Zeile

- `init-Programm?`
- `root-Device, rootflags, rootfstype, rootdelay, nfsroot?`
- `ip-Adresse`
- `boot-Eintrag`
- `resume-Aufruf, resume_offset, noresume`
- `quiet`
- `ro, rw`
- `debug`
- `break (top, modules, premount, mount, ...)` → **starten von Shell, reboot wenn `panic` eine Zahl** → `sleep`
- `blacklist` für Module

- Modulabhängigkeiten ermitteln: `depmod -a`
- Ausführen von `/scripts/init-top/`-Skripten:  
Festplatte ermitteln, Framebuffer initialisieren,  
Tastaturbelegung laden.
- `/scripts/init-premout` → Auswerten von  
**Modul-Blacklist**, Laden von Thermal-Modulen, Starten von  
`udev`
- Skript `/scripts/local` oder `/scripts/nfs` für  
Root-System laden

- Erstellen von `/dev/root`, **mounten** des Root-Systems
- Stoppen von `udev`, verschieben von `/dev`, `/sys` und `/proc` in das **neue** Root-Verzeichnis.
- Starten von `init`, entweder aus Boot-Zeile, `/sbin/init`, `/etc/init`, `/bin/init` oder sonst `/bin/sh` aus **neuem Root-Verzeichnis** (**chroot**)!

# Aktualisierung der initrd

- Erstellen und aktualisieren mit `update-initramfs` → Shellskript
- Konfigurationsdatei:  
`/etc/initramfs-tools/update-initramfs.conf`
- Kernelkonfiguration: `/etc/kernel-img.conf`
- ruft u.a. `mkinitramfs` auf → Shellskript
- Installation der `initramfs` als `initrd`
- Aktualisierung des **Bootloaders**

## Erstellen der initrd

- verwendet Verzeichnisse `/etc/initramfs-tools` und `/usr/share/initramfs-tools`:
  - `initramfs.conf` Konfiguration (module, busybox, etc)
    - `modules` Liste der Module die hinzugefügt werden sollen
    - `conf.d` Verzeichnis mit hartkodierten Bootparametern
- `cp -p /usr/share/initramfs-tools/init`  
`${DESTDIR}/init`
- Skripte aus `/usr/share/initramfs-tools/scripts/` und `/etc/initramfs/scripts`

# mkinitramfs

- `cp -p ${CONFDIR}/initramfs.conf  
${DESTDIR}/conf/`
- **kopieren der notwendigen Module**
- `copy_exec ${CONFDIR}/conf.d/* /conf/conf.d/`
- `copy_exec /sbin/modprobe /sbin`
- `copy_exec /sbin/rmmod /sbin`
- `cp -a /etc/modprobe.d/*  
${DESTDIR}/etc/modprobe.d/`
- `run_scripts  
/usr/share/initramfs-tools/hooks`
- `run_scripts ${CONFDIR}/hooks`
- **Aufruf von `cpio` und packen des erstellten Archivs**

# Das war es schon!