# Startup

Hans Buchmann FHNW/IME

19. November 2014

# Um was geht es ?

## wie startet ein Rechner

- am Beispiel **RaspberryPi**
- mit U-Boot
  - dazwischen

# Informationen

- `http://www.denx.de/wiki/U-Boot`

# Reset
## der Big-Bang

1. Reset Signal
2. Programmcounter pc bekommt einen Wert:
   - z.B. $pc \leftarrow 0$
3. der Code bei *pc* wird ausgeführt

# Reset
## beim **RaspberryPi**

1. Reset Signal
2. *first stage bootloader*
   - nicht zugänglich
3. *second stage bootloader* `bootcode.bin`
   - schwierig zugänglich
4. *GPU firmware* `start.elf`
   - **RaspberryPi** GPU Code
   - Konfiguration:
     - `config.txt`
     - `cmdline.txt` für GNU/Linux
   - ziemlich schwierig zugänglich
5. *User Code*
   - normalerweise GNU/Linux Kernel

# User Code
## GNU/Linux Startup

6. *kernel* `kernel.img`
7. UNIX `init` Prozess

# Aufgabe

## eigener UNIX `init` Prozess

- `init` ein normales UNIX Programm
- command line: `init=`*`myProcess`*

# Aufgabe

## U-Boot als zusätzlicher Zwischenschritt

6. U-Boot nun **kernel** startet
7. GNU/Linux **kernel**
8. UNIX

# Herstellung

- Code `git://github.com/swarren/u-boot.git`
- fast gleich wie der GNU/Linux **kernel**
- im File `config.txt` Eintrag:
  - `kernel=`*`name-of-uboot-image`*

# GNU/Linux
## mit U-Boot starten

- von der SD-Karte
- vom Netz
- ein U-Boot Image

# Etwas Terminologie

| | |
|---|---|
| direct-boot | **ohne** U-Boot |
| u-boot | **mit** U-Boot |
| kernel.img | das orignale *image* |
| Image | der selber gemachte |

Remark: Test mit `uname -a`

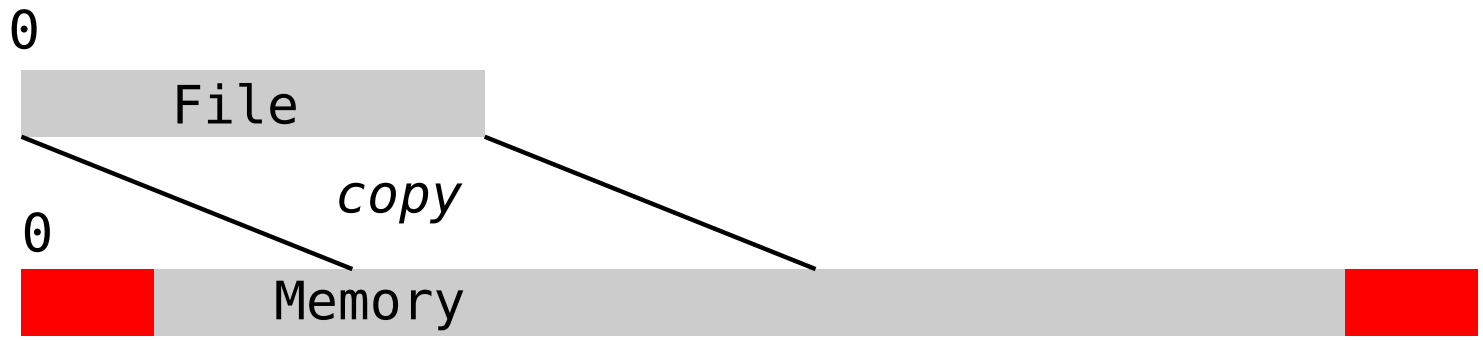
# Die Probleme

| | kernel.img | Image |
|---|---|---|
| direct-boot | ok | ok |
| u-boot | (ok) ohne rootfs | error |

## Aufgabe

- nachprüfen

# Memory
## U-Boot Sicht




- reserviert
- U-Boot `fatload mmc 0:1 addr file`

# Verschiedene Image Formate

- Image
- zImage
- uImage