

Ein kleines UNIX

Hans Buchmann FHNW/IME

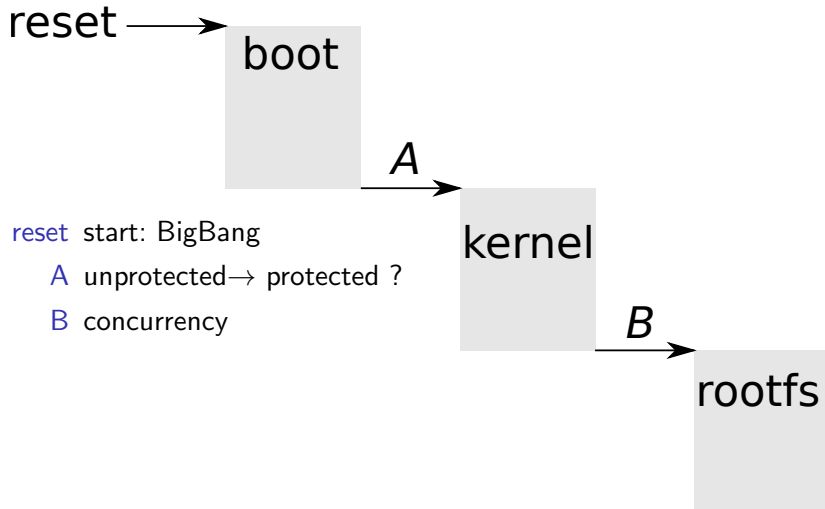
6. Dezember 2017

Ein kleines UNIX

- ▶ die einzelnen Komponenten
- ▶ für Test
- ▶ übersichtlich
- ▶ grosse Systeme haben ähnliche Komponenten

Die grossen Blöcke

Übergänge



Boot:u-boot

Zwei Files

reset

U-Boot SPL 2017.09 (Oct 24 2017 - 12:23:25)

- ▶ MLO Wegen TI
- ▶ u-boot.img

Kernel

Zwei Files

A

Booting Linux on physical CPU 0x0

Linux version 4.14.0-rc4+ (buchmann@buchmann) (gcc version 7.2.0 (GCC)) #1 SMP Tue Nov 21 09:35:03 CET 2017

- ▶ zImage *Der kernel*
- ▶ am335x-boneblack-wireless.dtb device tree

RootFS: Viele Files

Unser Interesse

A

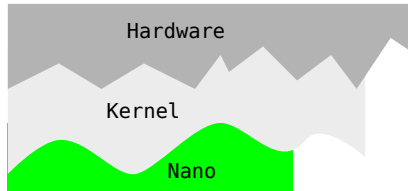
```
VFS: Mounted root (ext4 filesystem) on device 179:2.  
devtmpfs: mounted  
Freeing unused kernel memory: 1024K  
  
Please press Enter to activate this console.
```

- ▶ linuxrc Init-Process
- ▶ ...

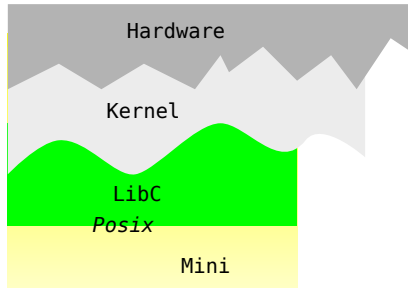
RootFS

Flavours

- ▶ nano
 - ▶ Assembler ohne *libraries* `s-nano.S`
 - ▶ **C** fast ohne *libraries* `c-nano.c`
- ▶ mini
 - ▶ *libraries*
 - ▶ static
 - ▶ dynamic
- ▶ full
 - ▶ busybox
 - ▶ ssh
 - ▶ ...



- ▶ `config/Makefile`
- ▶ `src/s-nano.S`
 `target-root` nichts
- ▶ `src/c-nano.c`
 `target-root` `libc.a` wegen `syscall`

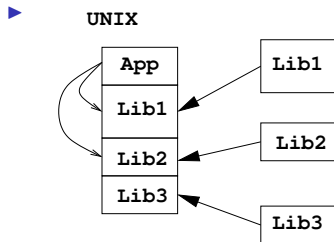


- ▶ `config/Makefile`
- ▶ `src/mini.c`
 - ▶ `static`
`target-root libc.a`
 - ▶ `dynamic`
`target-root libc.so, loader`

Statische/Dynamische Bibliothek

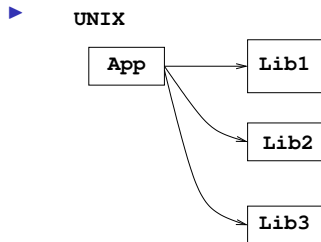
Kopie vs. Referenz

Static



► frühes Binden

Dynamic



► spätes Binden

—► copy
—> reference

Dynamische Bibliothek:der Loader

am Beispiel `mini.c`

- ▶ wir sind in `work`
- ▶ `gcc ../src/mini.c -o mini`
- ▶ `file mini`
- ▶ start loader
- ▶ start loader mit *executable*

Verzeichnisstruktur

19-minimal

```
├── config
│   └── Makefile
├── doc ..... die Folien
├── src ..... die source files
├── tc ..... toolchain normalerweise link
├── work ..... hier sind Sie
│   └── Makefile → ../config/Makefile
```

Die Files

`drive.switch.ch/index.php/s/SR9s26Wppx1Zvzq`

- ▶ `target-root-2017.12.06.tar.gz` minimales RootFS
- ▶ `gcc-7.2.0-arm-64bit-2017-12-06.tar.gz` volle Toolchain

Aufgaben

- ▶ Erzeugen Sie ein:
 - ▶ **s-nano**
 - ▶ **c-nano**
 - ▶ **mini-static**
 - ▶ **mini-dynamic**auf dem **BBB** RootFS
- ▶ ändern Sie den `Makefile` so ab, dass er **s-nano**, **c-nano**, **mini-static** und **mini-dynamic** auf den *Host* erzeugt