Devices revisited

Hans Buchmann FHNW/IME

13. März 2018

Um was geht es ?

C und Scripts

- Verbindung
  - *kernel-space* $\leftrightarrow$ *user-space*
- call-backs
  - verschiedene Formen
- Info `lxr.free-electrons.com`
- Schritt für Schritt mit `git`
  - `git log`
  - `gi show`

Um was geht es ?

- Kernel-Modules: **L**oadable **K**ernel **M**odule
  - `insmod`
  - `rmmodule`
- device
  - *major minor*
  - `devicefile` = *major minor*
- *kernel-space* $\leftrightarrow$ **devicefile** $\leftrightarrow$ *user-space*

Wie merkt der *kernel*-space

- ob ein **LKM**
  - eingefügt
  - entfernt

  wird

Wie merkt der *user*-space

- ob sich in einem **LKM**
  - etwas tut

  wird

Host	**BBB**
start	
`minicom -D/dev/ttyUSB0`	
	start
	`ifconfig usb0 192.168.7.7`
	`/sbin/sshd`
`ssh root@192.168.7.7`	
`sshfs root@192.168.7.7: mount`	

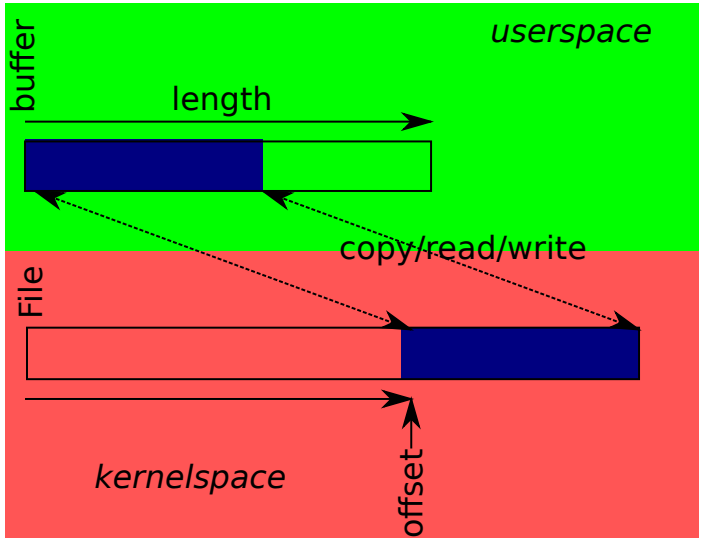
ready to develop

- init/exit
- `struct file_operations fops;`

empty read/write

- *kernel-space*
  - `file_operations`: read/write
    - mit `printk` call-back anzeigen
- *user-space*
  - `mknod device c` *`Major`* `0`
  - read: `cat device`
  - write: `echo abcd > device`

read/write



- read/write:
  - `copy_to_user`/`copy_from_user`
  - das Zusammenspiel:
    - `lenm*ofs`

Verbindung mit *userspace*: `/sys`

- *kernel-space*
  - `MODULE_LICENSE ("GPL")`
  - init: `simple_module=class_create(THIS_MODULE,"simple_device")`
  - exit: `class_destroy(simple_class)`
- *user-space*
  - `ls /sys/class`
- noch keine Informationen in `/sys/class/simple_device`

Verbindung mit *userspace*: `/sys`

- *kernel-space*
  - init: `device_create,KDEV(Major,0)`
  - exit: `device_destroy`
- *user-space*
  - `ls /sys/class/simple_device`