

Kernelspace-userspace call-back

Hans Buchmann FHNW/ISE

10. Dezember 2019

Outline

- ▶ The Big Picture
 - ▶ call-back, notification
- ▶ GNU/Linux KernelSpace-UserSpace am Beispiel **BBW**
 - ▶ hotplug
 - ▶ socket NETLINK KOBJECT UEVENT

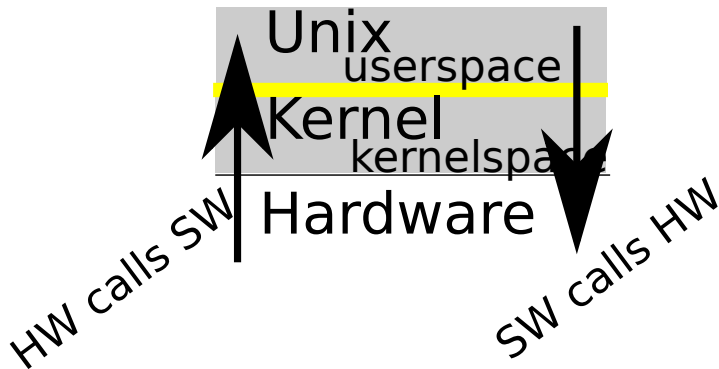
Remark: gilt für alle GNU/Linux

Typisches Szenario

USB-Stick einstecken

1. Hardware Interrupt
2. Kernel behandelt den Interrupt
3. Informiert (notifiziert) *user*
 - ▶ device name: `/dev/sdXY`
4. *user* montiert Filesystem
 - ▶ `mount /dev/sdXY /mount-path`

KernelSpace-UserSpace



SW calls HW

UserSpace to KernelSpace

- ▶ `date`
 - ▶ Klassisch der "normale" Fall

HW calls SW

KernelSpace to UserSpace

- ▶ Hardware bemerkt etwas USB-Stick
- ▶ Informiert den *user*
 - ▶ Klassisch der "abnormale" ungeliebte Fall

Informieren

verschiedene Wörter

- ▶ Notifizieren *notify*
- ▶ Zurückrufen *call-back*
- ▶ hotplug

Hotplug

/proc/sys/kernel/hotplug

- ▶ Kernel: *kernel-space* entdeckt ein neues Gerät
- ▶ Informiert (notify) User *user-space*: neues Gerät
 - ▶ call-back: /proc/sys/kernel/hotplug
 - ▶ enthält Name eines *user-space* executables

Remark: Kernel muss für *hotplug* konfiguriert sein

Hotplug Beispiel

- ▶ File: `hotplug.sh`
- ▶ executable
 - ▶ `chmod a+x hotplug.sh`
- ▶ Register
 - ▶ `echo absolute-path-to-hotplug.sh > /proc/sys/kernel/hotplug`

Socket

Alles ist ein Socket

- ▶ Endpunkt einer Verbindung (Buchse)
- ▶ verschiedene Typen
 - ▶ Internet
 - ▶ Verbindung vom kernel
 - ▶ `netstat`

Socket

KerneSpace-UserSpace

- ▶ Kernel: entdeckt neues Gerät (*device*)
- ▶ Kernel: sendet Daten an einen Socket
- ▶ User: liest die Daten

Beispiel

- ▶ C:
 - ▶ `uevent-userspace.c`
 - ▶ `split.c` für die schöne Ausgabe
- ▶ C++
 - ▶ `uevent.h/cc` das Modul
 - ▶ `uevent-demo.cc` eine einfache Anwendung