

# Linux Kernel

Anne Barsuhn

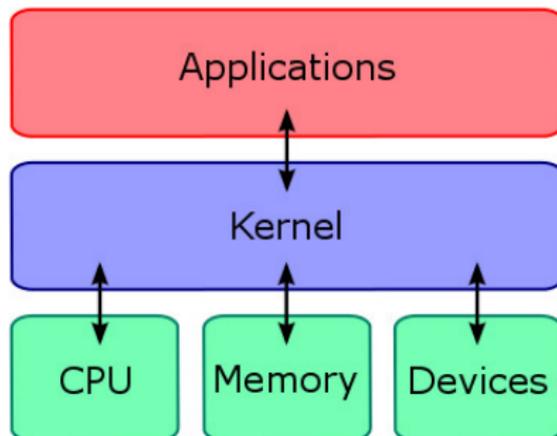
9. Februar 2013

## Aufbau eines Kernels

Was ist ein Kernel?

# Was ist ein Kernel?

- ▶ Grundbaustein eines Betriebssystems
- ▶ Kommunikation zwischen Hardware und Software (Programmen)



# Aufgaben eines Kernels

- ▶ teilt Programmen CPU-Rechenzeit zu (Round Robin)
- ▶ teilt Programmen Speicher zu
- ▶ lässt Programme mit USB- und anderen Geräten reden
- ▶ ermöglicht Programmen untereinander zu kommunizieren

## Wo ist mein Modul?

- ▶ `insmod [options] filename [args]`
- ▶ `insmod -V first_module.ko mystring="hackspace_jena"`
- ▶ gibt Versionsnummer auf dem Terminal aus

# Wie fine ich mein Modul?

- ▶ lsmod

- ▶ 

Module	Size	Used by
first_module	12364	0
nls_utf	812416	0

- ▶ Module - Name des Moduls

- ▶ Size - Größe des Moduls in Bytes

- ▶ Used by/Usecount - wie viele Geräte/gemountete Dateisysteme von dem geladenen Modul abhängen

- ▶ Zum Ausprobieren: `cat /proc/modules` oder `ls /sys/module`

# So baut man Kernelmodule

- ▶ Ziemlich einfach, oder?
- ▶ Was steckt dahinter und was kann man Cooles damit machen?
  - ▶ Ziemlich viel.
  - ▶ Das Betriebssystem gehört dir. :)

# Aufbau eines Linux Betriebssystem

- ▶ Everything is a file!!!! <sup>1</sup>
- ▶ `ls /`
- ▶ Wichtig für uns:
  - ▶ `/proc`
  - ▶ `/sys`
  - ▶ `/dev`
  - ▶ `/var` (schon benutzt)

---

<sup>1</sup><http://www.tldp.org/LDP/Linux-Filesystem-Hierarchy/html/index.html>

# Devices/Geräte

- ▶ `/dev` listet die devices/Geräte
- ▶ z.B. `/dev/urandom` oder `/dev/random` für random number generator in der Kryptographie
- ▶ z.B. `/dev/dsp` oder `/dev/audio` ist das Audio device
- ▶ Hör dir den Kernel an:
  - ▶ `sudo cat /vmlinuz > /dev/dsp` oder `sudo cat /dev/urandom > /dev/audio`
- ▶ Wenn `dsp` und `audio` nicht gelistet:
  - ▶ Debian:
    - ▶ `sudo aptitude install oss-compat`
    - ▶ `sudo modprobe snd-pcm-oss`
- ▶ `/dev/dsp` ist schon veraltet und wird auf Fedora so nicht mehr umgesetzt, dort gibt es `/dev/snd/`

# Devices/Geräte

- ▶ es gibt character devices und block devices
- ▶ Character devices sind für den Datentransfer zuständig
  - ▶ z.B. für die Kommunikation zwischen Audioplayer und Lautsprechern
  - ▶ Audioplayer schreibt nach `/dev/audio` und Lautsprecher lesen aus `/dev/audio` und geben wieder
- ▶ Block devices halten Daten z.B. Festplatten und CD-Laufwerke
- ▶ `cat /proc/devices`
- ▶ Wir bauen jetzt ein Charakter Device:

# Major/Minor Numbers

- ▶ `ls -l /dev`
- ▶ permission, links, author, user, major nr., minor nr., date, time, name
- ▶ major number - welcher Gerätetreiber für ein das Gerät benutzt werden soll
- ▶ minor number - ist ein Flag für den Gerätetreiber
- ▶ Was sind die Major/Minor Numbers von `urandom`, `random` und `null`?

## Wie erzeuge ich ein device?

- ▶ `sudo mknod testdev c 1 9`
- ▶ Auslesen des devices:
- ▶ `hexdump testdev`
- ▶ `< testdev tr -dc [:print:] — head -c 10 > test.txt`

# ENDE

- ▶ Vielen Dank, dass ihr mitgemacht habt
- ▶ Wenn Interesse an einer Weiterfuehrung besteht, ich hab noch mehr zu erzaehlen!
- ▶ z.B. herausfinden, welcher Virtuelle Speicher auf welchen Physischen Speicher gemapped wird und vieles mehr.
- ▶ Zum Nachschauen: <http://lxr.free-electrons.com/>