

# UNIX use

Hans Buchmann FHNW/ISE

24. September 2019

# Ziel

## Entwicklung von Programmen auf dem **BeagleBoneWireless**

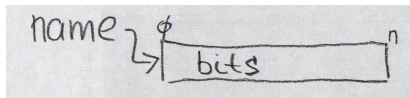
- ▶ alles ist ein File
    - ▶ 0 – *te* Näherung
    - ▶ File: *stream of bits*
  - ▶ Filesysteme
    - ▶ mount
    - ▶ sshfs
  - ▶ Cross development
    - ▶ *Host* ↔ **BeagleBoneWireless**
- Remark: Keine Toolchain auf dem **BeagleBoneWireless**

# Wichtig

- ▶ wo ist was ?
  - ▶ Verzeichnisstruktur
- ▶ wo sind wir ?
  - ▶ *Host*  
oder
  - ▶ **BeagleBoneWireless**

# Alles ist ein File

stream of bits



- ▶ *name* Referenz auf die Bits (Bytes)
  - ▶ Bits(bytes) der Reihe nach
  - ▶ indexiert  $0 \dots n - 1$
- ▶ File
  - ▶ Datenquelle
    - ▶ liefert Daten: Bits(Bytes)
  - ▶ Datensenke
    - ▶ absorbiert Daten: Bits(Bytes)

# Ein paar Befehle

- ▶ `cat name`
  - ▶ *concatenate files and print on the standard output*
  - ▶ zeigt den Inhalt
- ▶ `hexdump -C name`
  - ▶ *display file contents in hexadecimal, decimal, octal, or ascii*
  - ▶ zeigt die Bits hexadezimal
- ▶ `dd if=... of=... count=...`
  - ▶ *convert and copy a file*
  - ▶ brauchen wir oft

# Devices sind auch Files

## am Beispiel SD-Karte

`/dev/mmcblki`  $i = 0, 1, 2 \dots$

**Remark:** Name vom Betriebssystem bestimmt

**Datenquelle** `hexdump -C /dev/mmcblk0`

**Datensenke** `cp name /dev/mmcblk0`

**Remark:** Aufpassen

## Devices sind auch Files

z.B Zufallszahlen

`/dev/random` sammelt das Rauschen: langsam

**Remark:** Name vom Betriebssystem bestimmt

**Datenquelle** `hexdump -C /dev/random`

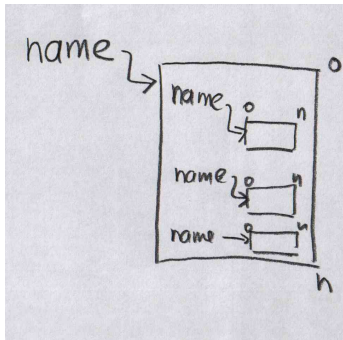
`/dev/urandom` berechnete (Pseudo) Zahlen: schnell

**Datenquelle** `play -b 16 -e signed-integer \  
-t raw -r 44000 /dev/urandom`

**Remark:** der Befehl `play` hat viele Optionen

# Filesystem

## Files für Files



► File der weitere Files enthält

► Verschiedene Filesysteme

`vfat` Microsoft

`ext4` UNIX

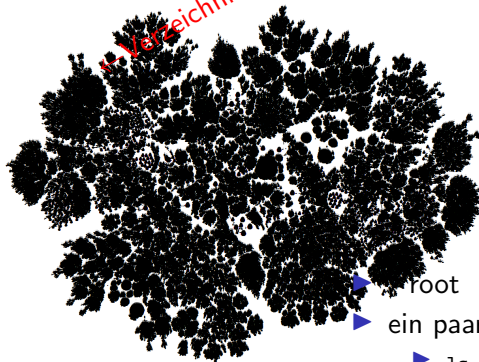
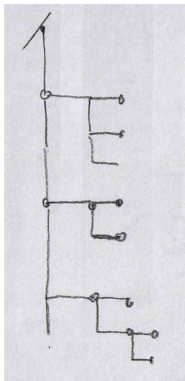
... noch viele andere

`cat /proc/filesystems`



# Vereichnisstruktur

## Hierarchie

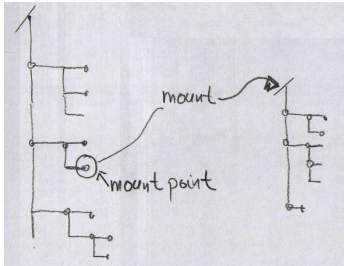


Verzeichnisse einer Workstation

- ▶ root
- ▶ ein paar Befehle
- ▶ `ls`, `tree`, `cd`

```
mount fileSystem mountPoint
```

Verbindet Filesysteme

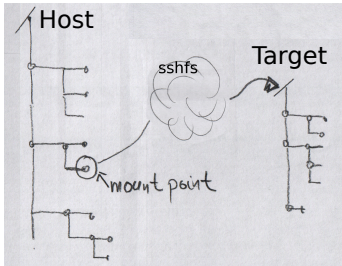


▶ `mount /dev/mmcblk0p1 mountPoint`

▶ `umount mountPoint`

**Remark:** *mountPoint* sieht wie ein normales Verzeichnis aus

`sshfs user@target:path mountPoint`  
via ssh



- ▶ `sshfs user@target:path mountPoint`
  - ▶ braucht ssh
- ▶ `fusermount -u mountPoint`
  - ▶ `umount`

**Remark:** *mountPoint* Sieht wie ein normales Verzeichnis aus