

UNIX use

Hans Buchmann FHNW/IME

22. September 2015

Ziel

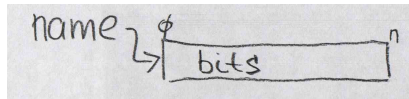
Entwicklung von Programmen auf dem **BeagleBoneBlack**

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

Wichtig

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

Alles ist ein File



- ▶ *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`
 - ▶ zeigt den Inhalt
- ▶ `hexdump -C name`
 - ▶ zeigt die Bits hexadezimal

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`

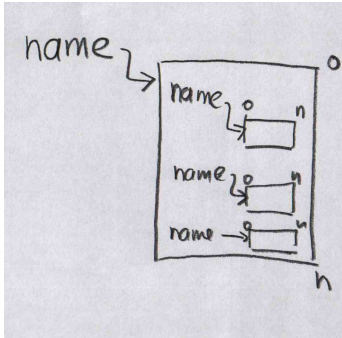
`/dev/urandom` berechnete (Pseudo) Zahlen: schnell

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

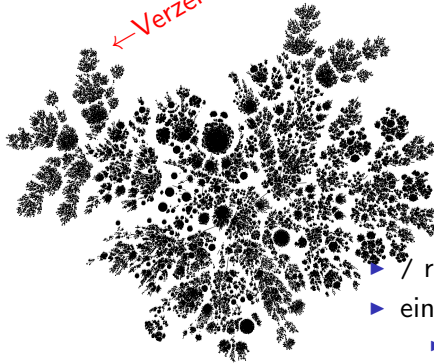
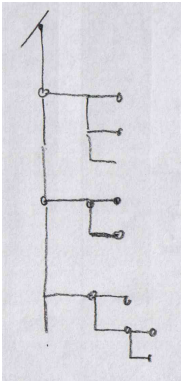
... noch viele andere

`cat /proc/filesystems`

Vereichnisstruktur

Hierarchie

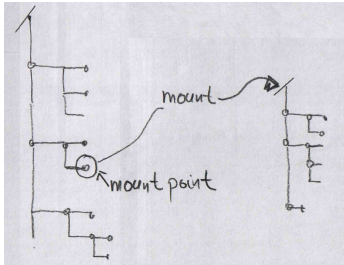
← Verzeichnisse einer Workstation



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

`mount fileSystem mountPoint`

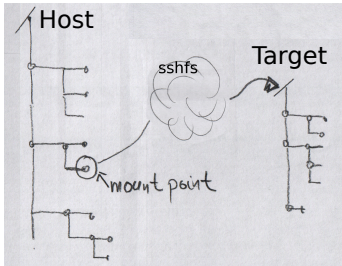
Verbindet Filesysteme



- ▶ `mount /dev/mmcblkOp1 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

Verzeichnisstruktur

Host ↔ **BeagleBoneBlack**

Host

```
devel ..... somewhere on the host
├── config
│   └── Makefile  for making BeagleBoneBlack executables
├── src .....c,c++
├── tc ..... normally toolchain
└── work ..... connected with BeagleBoneBlack current dir
```

BeagleBoneBlack

```
user ..... somewhere on the BeagleBoneBlack
└── work ..... connected with Hostcurrent dir
```

Vorbereitung

- ▶ Verbindung mit **BeagleBoneBlack** via ssh
- ▶ Installation *toolchain* auf *Host*

Verbindung mit **BeagleBoneBlack**

- ▶ `ssh` für die Ausführung der Programme
- ▶ *mount* Varianten
 - ▶ *Host* auf **BeagleBoneBlack**
 - ▶ **BeagleBoneBlack** auf *Host*
- ▶ Kopiere *executable* auf **BeagleBoneBlack**
 - ▶ `scp` secure copy
 - ▶ `scp executable user@target:`

Programme auf BeagleBoneBlack

- ▶ C Programme: `hello-world-c.c`
- ▶ C++ Programme `hello-world-cpp.cc`, `primes.cc`

Remark: Toolchain funktioniert für C++ noch nicht