

Debugging

Hans Buchmann FHNW/IME

17. März 2015

Dokumentation

- ▶ Gemacht mit `sh tools/make-it.sh htmldocs in 5-kernel`
- ▶ Daten in `build/Documentation/DocBook/index.html`

Um was geht es ?

Fehlersuche

- ▶ im *userspace* normale Programme
 - ▶ *Host*
 - ▶ **RaspberryPi**
- ▶ im *kernelspace*
 - ▶ *Host*
 - ▶ **RaspberryPi**

printf/fprintf

- ▶ Unterschied stdout vs. stderr
 - ▶ `printf (stdout)`
 - ▶ `fprintf(stderr,....)`
- ▶ Einfaches Beispiel `hello-world.c`

strace

trace system calls and signals

▶ `strace ./hello-world`

Remark: Einen Haufen Optionen

Debugger

sourceware.org/gdb/current/onlinedocs/gdb/

- ▶ compile -g
- ▶ start: `gdb -d path-to-source`
- ▶ break point
- ▶ run, step, cont

die graphische Oberfläche

z.B. nemiver

- ▶ break point
- ▶ run cont
- ▶ etc.

Remark: es gibt noch andere graphische Oberflächen

TODO's

- ▶ Optimierungen
 - ▶ Makefile CFLAGS -O0, -O1, -O2

Files

► dmesg

printk

Das *kernel printf*

- ▶ Siehe `6-modules/src/simple-module.c`

strace

► strace gibt es natürlich nicht

Zwei Rechner

Host Target

	Host		Target	
seriell	/dev/ttyUSB0	↔	/dev/ttyS	debug
ethernet	eth0	↔	eth0	Bedienung

Remark: die serielle Schnittstelle kann nicht für die Bedienung gebraucht werden

Konfiguration

Target Kernel hacking → KGDB kernel debugger

Host gdb für ARM

download www.gnu.org/software/gdb/
configuration `configure --prefix=where-on-host \
--target=arm-none-linux-gnueabi`

Anwendung

- ▶ Siehe `DocBook/index.html kgdb`