

Admin

Hans Buchmann FHNW/IME

12. September 2014

Folien/Code `websvn.fhnw.ch/trac/edu/browser/linux-lab`
Prüfung mündliche MSP

- ▶ Alles öffentlich zugängliche Material zu dieser Vorlesung unterliegt der *GNU GENERAL PUBLIC LICENSE*, auch wenn das in den einzelnen Dokumenten nicht explizit angegeben ist.
- ▶ <http://www.gnu.org/copyleft/gpl.html>

- ▶ GNU/Linux from Scratch
- ▶ Einblick in die Mechanismen
- ▶ Umgang mit verschiedenen Tools
- ▶ Weniger programmieren, mehr konfigurieren
- ▶ Schrittweises Vorgehen: (fast) immer lauffähiges System

- ▶ Führen Sie ein Laborbuch bzw. Laborfile

- ▶ Ausleihe
- ▶ Spannung 4.5-6 Volt: **nicht mehr**

The Big Picture

—sourcefile— —revision— —time— —owner—

Hans Buchmann FHNW/IME

12. September 2014

- ▶ GNU/Linux ist:
 - ▶ Software mit klassischen Methoden hergestellt
 - ▶ gross
 - ▶ komplex nicht kompliziert
- ▶ Darum:
 - ▶ Die grundlegenden Mechanismen beachten
 - ▶ Übersicht bewahren
 - ▶ Verzeichnisstrukturen: wo ist was.

- ▶ $\approx 10M$ SLOC (Source Lines of Code)
- ▶ $\approx 2.3K$ Verzeichnisse
- ▶ $\approx 33K$ Files davon
 - ▶ $\approx 30K$ $\{c|h\}$ -Files
 - ▶ $\approx 1K$ Assembler Files
 - ▶ $\approx 1.4K$ Makefiles
 - ▶ Rest: Makefile, Scripts etc.

Remark(s):

- ▶ $M = 10^6$ $K = 10^3$
- ▶ Gemacht mit `sloccount`

C Unabhängig von Rechnerarchitektur, Hauptsprache für
Bootloader, Kernel, libc

Assembler Für kleine Anpassungen

Skript Für Routineaufgaben

Makefile Für den Zusammenbau

Compiler gcc gcc.gnu.org

binutils Sammlung von Programmen¹
(www.gnu.org/software/binutils)

Assembler as

Linker ld

Maker make www.gnu.org/software/make

¹Liste nicht vollständig

BootLoader *reset* Handler, SingleUser

Kernel Prozessverwaltung, Treibersammlung

libc Normierte (POSIX) Schnittstelle, Kernel-UNIX

UNIX Filesystem, Sammlung von Programmen und Daten

Komponenten lassen sich:

- ▶ einzeln herstellen
- ▶ kombinieren
- ▶ austauschen

BootLoader nicht flüchtiger Speicher: z.B. Flash

Kernel RAM

libc RAM

UNIX RAM,Harddisk,MemoryCard,NFS

- [openEmbedded](http://www.openembedded.org) www.openembedded.org
(á la Gentoo) Skriptsammlung/Repository, für
viele Rechnerarchitekturen
- [buildroot](http://buildroot.uclibc.org) buildroot.uclibc.org
Makefiles/patches für viele Rechnerarchitekturen
- [CLFS](http://cross-lfs.org/view/clfs-embedded/arm) cross-lfs.org/view/clfs-embedded/arm
Cross-**C**ompiled **L**inux **F**rom **S**cratch
Anleitung

- openEmbedded/buildroot ▶ Grosse Systeme
 - ▶ Braucht zusätzliche *tools*
 - ▶ Führen zusätzliche Komplikationen ein
- CLFS Sehr rezeptartiger Aufbau

Warum ?

- ▶ Basiert auf der originalen Software
- ▶ Die klassischen Tools (`Makefile`) sind schon sehr gut ausgebaut.
- ▶ Brauchen tieferen Einblick in das ganze System
- ▶ Nur ein bis zwei Rechnerarchitekturen

0-intro Diese Folien

1-tools Die Werkzeuge

2-unix-use UNIX aus Benutzersicht: Host und Target

3-uboot Wie startet ein Rechner

4-kernel Das GNU/Linux : Konfiguration/Herstellung

5-libc Verbindung GNU/Linux UNIX

6-unix UNIX Konfiguration/Herstellung

7-build ein *build* System

Entwicklungsumgebung

—sourcefile— —revision— —time— —owner—

Hans Buchmann FHNW/IME

12. September 2014

Host Entwicklungsrechner, GNU/Linux Betriebssystem

Target *SAM9-L9260*

RS232 u-boot *shell*, GNU/Linux console

Ethernet IP, TFTP etc. IP Stack

MemoryCard u-boot, Kernel, UNIX

Editor ASCII Editor,joe,nedit etc. etc.

Compiler gcc

mkimage Erzeugt u-boot Image

Terminal minicom ev. USB ↔ RS232 Kabel

UNIX Basibefehle ls,cp, etc.

Codeverwaltung svn cvs git etc.

Eclipse kommt später