

# *yocto* GNU/Linux nach Mass ?

Hans Buchmann FHNW/IME

7. Januar 2015

## Um was geht es ?

- ▶ Massgeschneidertes GNU/Linux für:
  - ▶ heterogene Hardwareplattformen
  - ▶ eingebettete Systeme
- ▶ mit
  - ▶ standardisierter (einfacher ?) Herstellung

# The Big Picture

## GNU/Linux & Co.

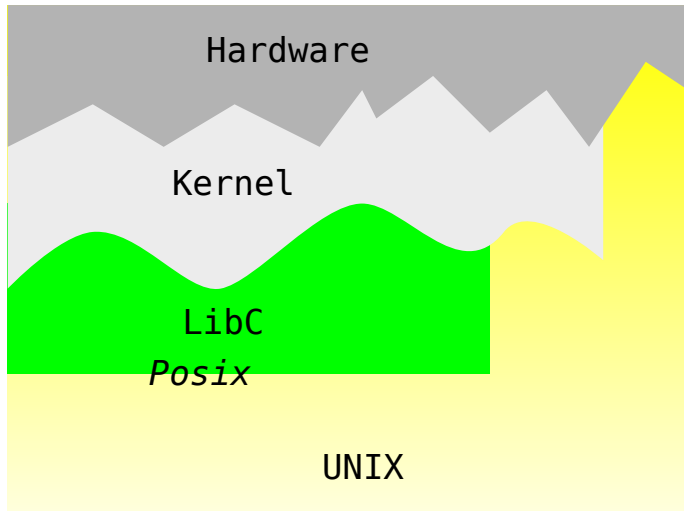
### einerseits

- ▶ besteht GNU/Linux aus nur ein paar Komponenten
- ▶ ist GNU/Linux relativ modern

### andererseits

- ▶ ist GNU/Linux sehr komplex
- ▶ ist GNU/Linux klassisch hergestellt
  - ▶ mit über 30-jährigen Konzepten

# Die Komponenten layers



## Die Komponenten

HW heterogen:

- ▶ Architekturen ARM , *Intel*
- ▶ Peripherie

Kernel das eigentliche GNU/Linux

- ▶ Sammlung von *drivern*
- ▶ Scheduling
- ▶ Verwaltung der Ressourcen

LibC die POSIX Norm

- ▶ <http://pubs.opengroup.org/onlinepubs/9699919799/>

UNIX der Rest:

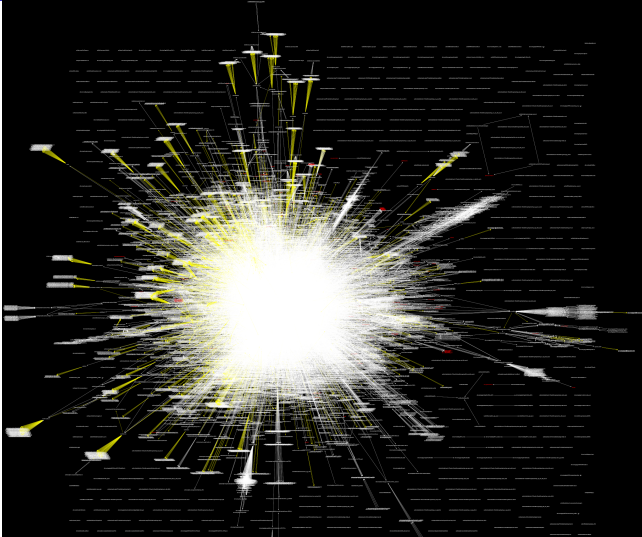
- ▶ alles ist ein File

# Die Komplexität

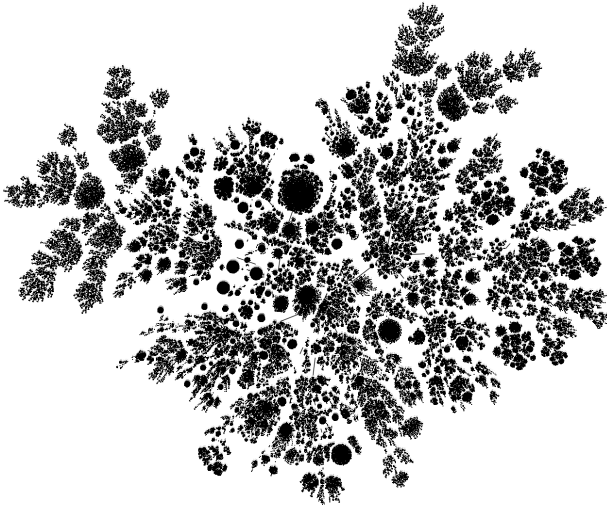
## Zwei Beispiele

- ▶ die *include* Relationen im *kernel*
- ▶ die Verzeichnisstruktur einer UNIX Workstation

## Include Relationen nur im **kernel**



# Verzeichnisstruktur einer Workstation





# Die Herstellung die klassischen Methoden

## C die dominante Programmiersprache

### make Steuerung der Herstellung

Copyright © 1978 by Bell Telephone Laboratories, Incorporated.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. Printed in the United States of America. Published simultaneously in Canada.

This book was set in Times Roman and Courier 12 by the authors, using a Graphic Systems phototypesetter driven by a PDP-11/70 running under the UNIX operating system.

UNIX is a Trademark of Bell Laboratories.

# The Big Picture

## Die Herstellung

### Gegeben

- ▶ eine Hardware
- ▶ korrekte<sup>1</sup> Sourcefiles
  - ▶ hauptsächlich **C**
  - ▶ verstreut auf der Welt
  - ▶ selber geschrieben

### Gesucht

- ▶ ein Produkt
  - ▶ z.B. ein GNU/Linux basiertes eingebettetes System

### Lösungsweg

- ▶ z.B. *yocto*
- ▶ oder etwas anderes

---

<sup>1</sup>meistens

## Das Problem

die richtige Wahl

- ▶ die **richtigen** Sourcefiles
- ▶ **richtig** konfiguriert

yocto

<https://www.yoctoproject.org>

## Gegeben

- ▶ eine Hardware
- ▶ korrekte<sup>2</sup> Sourcefiles
  - ▶ hauptsächlich **C**
  - ▶ verstreut auf der Welt
  - ▶ selber geschrieben

## Gesucht

- ▶ ein Produkt
  - ▶ z.B. ein GNU/Linux basiertes eingebettetes System

## Lösungsweg

- ▶ bitbake *theProduct*

---

<sup>2</sup>meistens

## Wichtige Begriffe

- ▶ Package
  - ▶ Abhängigkeiten
- ▶ Konfiguration
  - ▶ die richtige Wahl
- ▶ Versionen
  - ▶ z.B. (und vor allem) `git`
- ▶ Image
  - ▶ das Endprodukt

## Wichtige Begriffe

- ▶ Layer
  - ▶ die Verzeichnisse `meta-*`
- ▶ SDK Software Development Kit
  - ▶ die *toolchain*
- ▶ bitbake
  - ▶ das Arbeitspferd
- ▶ Rezept *recipe*
  - ▶ beschreibt die Herstellung einer Komponente

## Vorbereitung

[www.yoctoproject.org/docs/1.7/yocto-project-qs/yocto-project-qs.html](http://www.yoctoproject.org/docs/1.7/yocto-project-qs/yocto-project-qs.html)

- ▶ Installation
  - ▶ → Getting the Yocto Project
- ▶ Setze korrekte Umgebung
  - ▶ `. oe-init-build-env`
- ▶ Starte
  - ▶ `bitbake -k core-image-minimal`
  - ▶ braucht Zeit
  - ▶ braucht  $\approx 40GiB$  Festplatte

# Das Resultat als virtuelle Machine

Die Files:

**kernel**

`bzImage-qemux86.bin`

**UNIX**

`core-image-minimal-dev-qemux86-20141018183055.rootfs.ext3`

**Starten**

`runqemu qemux86 kernel unix serial`

**Remark:** predefined:

<http://downloads.yoctoproject.org/releases/yocto/yocto-1.7/machines/qemu/qemux86/>



## *yocto* einrichten

- ▶ <https://www.yoctoproject.org/downloads>
- ▶ <http://www.yoctoproject.org/docs/1.7/yocto-project-qs/yocto-project-qs.html>

## yocto Predefined

- ▶ <http://www.yoctoproject.org/docs/1.7/yocto-project-qs/yocto-project-qs.html>
  - ▶ Using Pre-Built Binaries and QEMU
- ▶ die Files:
  - ▶ bzImage-qemux86.bin **kernel**
  - ▶ core-image-minimal-dev-qemux86-\*.rootfs.\* rootfs
- ▶ qemu-system-i386 die Parameter:
  - ▶ -kernel bzImage-qemux86.bin
  - ▶ -append "root=/dev/hda console=ttyS0"

*yocto*

Erstes Target

- ▶ bitbake core-image-minimal