

Das Yocto Projekt

06.06.2019

Fabian Krüger & Michael Scheiderer

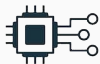
FAU Informatik, Lehrstuhl 4

- Yocto - Ein Überblick
- Organisation
- Die technische Seite



Yocto - Ein Überblick

*THE YOCTO PROJECT.
IT'S NOT AN EMBEDDED LINUX DISTRIBUTION,
IT CREATES A CUSTOM ONE FOR YOU.*



EMBEDDED SYSTEM

shutterstock.com • 1192195141

+



=

yocto .
PROJECT

- Mit Yocto werden Embedded-Linux-Systeme erstellt
- Update/Änderungsprozess schnell und automatisierbar
- *Toolchain*, die für die Erstellung gewährleistet:
 - schnell
 - standardisiert
 - stark spezialisiert

Die Ziele des Yocto-Projekts

- Plattform für einheitliche Entwicklung von Embedded-Linux-Systemen
- Standardisierung und Wiederverwendung von bereits existenten Linux-Komponenten (z.B. Kernel, Treiber, ...)
- Erstellungsprozess bestmöglich unabhängig von Zielhardware gestalten
- ...

Alle Ziele für Interessierte:

<https://www.yoctoproject.org/about/charter/>

Organisation

- hierarchischer Aufbau
 - Ein Projekt-Architekt (aktuell: Richard Purdie)
 - Hauptprojekt setzt sich auf oberster Ebene aus allen Teilprojekten zusammen
 - Jedes Teilprojekt wird von eigener Instanz betreut
- Positionen im Projekt durch Anzahl und Qualität der Beiträge
 - zum Beispiel Leitung eines Teilprojekts

Memberships

- (kommerzielle) Organisationen:
 - viele Arbeitsanteile
 - finanzielle Beteiligung
 - Teil der Projektverwaltung
 - Ratgeber:
Wünsche der Industrie an Yocto
 - unterteilt in verschiedene Kategorien
 - Platin
 - Gold
 - Silber
- Kategorien haben
Abzeichenfunktion



Beispiel für Member:

- Intel
- AMD
- Facebook
- Openembedded
- ...

Die technische Seite

Änderungen und Releases

Neue Yocto-Versionen:

Codename	Version	Release	Aktuelle Version
Zeus	2.8	Oktober 2019	
Warrior	2.7	April 2019	2.7.0
Thud	2.6	November 2018	2.6.2
Sumo	2.5	April 2018	2.5.3
Rocko	2.4	Oktober 2017	2.4.4
Pyro	2.3	Mai 2017	2.3.4
⋮	⋮	⋮	⋮
Inky	?	c.a. 2007	?

- regelmäßig neue Versionen
- fester Zeitplan für Releases
- dauerhafte Unterstützung der meisten Versionen

Linux-Kernel 5.0.17 vs. neueste Version Yocto-Kernel

```
diff -urN linux-5.0.17/{...}/trace_events_hist.c linux-yocto_git/...
--- linux-5.0.17/{...}/trace_events_hist.c      2019-05-16 ...
+++ linux-yocto_git/{...}/trace_events_hist.c    2019-05-19 ...
@@ -4695,10 +4695,9 @@
         /* ensure NULL-termination */
         if (size > key_field->size - 1)
             size = key_field->size - 1;
+    }

-    strncpy(compound_key + key_field->offset, (char *)key, size);
-    } else
-        memcpy(compound_key + key_field->offset, key, size);
+    memcpy(compound_key + key_field->offset, key, size);
 }
```

- Kaum Unterschiede → sehr Ähnlich zum Linux-Kernel
- meist kleine Änderungen, z.B. elegantere Lösungen für Problem

Für das erste eigene Projekt mit Yocto wichtig:

- **Layer:**

- Metadaten für einen Build / Teil des Builds
- Unterstützen die Modularisierung des Builds

- **Recipes:**

- kleinste Bausteine eines Builds
- Downloadpfade & Compileanweisungen

- **Bitbake:**

- Benutzt Layer und ihre Recipes
- Anweisungen der Layer + holen und compilieren der Ressourcen der Recipes = Build des Images

Layers, Recipes & Erweiterungen

Layer - Anweisungen & Metadaten für einen build:

```
# We have a conf and classes directory, add to BBPATH
BBPATH .= ":{LAYERDIR}"
```

```
# We have recipes-* directories, add to BBFILES
BBFILES += "${LAYERDIR}/recipes-*/**/*.bb \
           ${LAYERDIR}/recipes-*/**/*.bbappend"
```

```
BBFILE_COLLECTIONS += "yoctobsp"
BBFILE_PATTERN_yoctobsp = "^${LAYERDIR}/"
BBFILE_PRIORITY_yoctobsp = "5"
LAYERVERSION_yoctobsp = "4"
LAYERSERIES_COMPAT_yoctobsp = "warrior"
```

Quelle: Auszug aus meta-yocto-bsp layer.conf

Recipes: Sourcepfade und Konfigurationen für Packages

```
{...}  
SRC_URI = "git://git.yoctoproject.org/linux-yocto.git;\ \  
          branch=${KBRANCH};name=machine \  
          git://git.yoctoproject.org/yocto-kernel-cache;\ \  
          type=kmeta;name=meta;branch=\ \  
          yocto-5.0;destsuffix=${KMETA}"  
LINUX_VERSION ?= "5.0.7"  
{...}  
KERNEL_FEATURES_append_qemuall=" cfg/virtio.scc"  
KERNEL_FEATURES_append_qemux86=" cfg/sound.scc cfg/paravirt_kvm.scc"  
KERNEL_FEATURES_append_qemux86-64=" cfg/sound.scc cfg/\ \  
                                     paravirt_kvm.scc"
```

Quelle: Auszug aus linux-yocto-rt_5.0.bb

- Recipes schnell erstellbar
- auch von verschiedenen Layers verwendbar

Baue ein Embedded-Linux:

1. Clone das Repository:

```
git clone git://git.yoctoproject.org/poky
git checkout tags/yocto-2.7 -b yocto-passt-schon-2.7
```

2. Erschaffe Ordnerstruktur, setze Umgebungsvariablen:

```
source oe-init-build-env
```

3. Kurzes checken der buildconfig (build/local.conf):

```
MACHINE ??= "qemux86"
# Default policy config
DISTRO ?= "poky"
{...}
# CONF_VERSION is increased each time build/conf/ changes
CONF_VERSION = "1"
```

4. Image-Build mittels bitbake

```
bitbake core-image-sato
```


Erweitere das Embedded-Linux: z.B. Image nach Build ssh-fähig machen:

1. erweitere Layer um ssh-Support:

```
# ssh-support  
CORE_IMAGE_EXTRA_INSTALL += "openssh"
```

2. neu bauen:

```
bitbake core-image-minimal  
{...}  
TARGET_SYS           = "i686-poky-linux"  
MACHINE              = "quemux86"  
DISTRO               = "poky"  
DISTRO_VERSION       = "2.7+snapshot-20190525"  
{...}
```

3. Image tatsächlich um Funktionalität erweitert:

```
./core-image-minimal/1.0-r0/rootfs/etc/init.d/sshd  
./core-image-minimal/1.0-r0/rootfs/usr/sbin/sshd
```

Erfüllt Yocto seine Ziele / seinen Zweck?

- standardisierte & anpassbare Erstellung ✓
- schnelle Änderungsprozesse ✓
- einsteigerfreundlich (nach Selbsttest) ✓

⇒ Yocto ist ein umfangreiches, fortschrittliches Werkzeug und gut geeignet um Embedded-Linux-Systeme zu erstellen und zu verwalten.

Vielen Dank für Eure
Aufmerksamkeit!

Fragen...?

Bonus: Participants/Compatible

Company (= Participant)	Product (= Compatible)	YP-Release	Approved
MontaVista Software	MontaVista Carrier	2.2	5/12/2017
Renesas	Renesas R-Car Gen3	2.1	5/12/2017
⋮	⋮	⋮	⋮



Quelle: <https://www.yoctoproject.org/ecosystem/branding/>

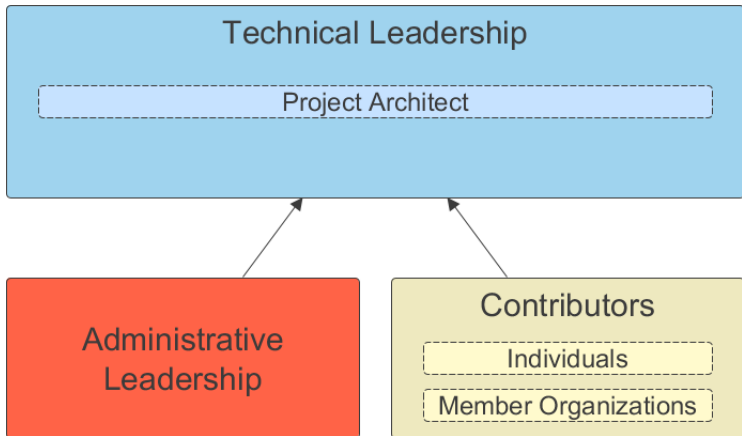
Bonus: Community

- eher klein, verglichen mit großen Distributionen wie z.B. Ubuntu
- → Einheitlichere Kommunikationskanäle
- → Guter Umgangston untereinander

fray	last time I was there I drove by it a few times on the way to where I'm going.. ;)
fray	but thats been as close as I've gotten.. ;)
eyalgal	fray: thanks
LetoThe2nd	fray: its really not super important. but nice if you want to be tourist for an hour, or so.
LetoThe2nd	fray: and, drink all the beers that are way too expensive in finland

Quelle: Yocto-IRC, 15.05.2019

Grobschema: Aufbau des Yocto-Projekts



Layers, Recipes & Erweiterungen

Layer - eine Sammlung von Recipes:

Beispiel: `tree -L 2 -charset ./recipe*` im Layer-Verzeichnis:

```
./recipes-graphics
-- xorg-xserver
  |-- xserver-xf86-config
  -- xserver-xf86-config_0.1.bbappend
./recipes-kernel
-- linux
  |-- linux-yocto_4.19.bbappend
  |-- linux-yocto_5.0.bbappend
  -- linux-yocto-dev.bbappend
```

- leicht erweiterbar
- (teilweise) wiederverwendbar → *overriding*