Санкт-Петербургский политехнический университет

Институт Компьютерных Наук и Технологий

Кафедра компьютерных систем и программных технологий

Лабораторная работа по дисциплине «Проектирование ОС и компонентов» на тему:

Платформа сборки дистрибутивов Linux Для встраиваемых устройств

«yocto»

Выполнил: студент группы № 63501/3

Дедков С.В.

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Душутина Е.В.

Санкт-Петербург

2016

# Оглавление

[Оглавление 2](#_Toc445786930)

[Цель работы 3](#_Toc445786931)

[Описание проекта, инструментов и возможностей 4](#_Toc445786932)

[Описание процесса сборки дистрибутива 6](#_Toc445786933)

[Процесс сборки с использованием web оболочки Toaster 8](#_Toc445786934)

[Вывод 11](#_Toc445786935)

[Список литературы 12](#_Toc445786936)

# Цель работы

* Изучить предоставляемые *Yocto* *project* шаблоны, инструменты и методы для создания специальных дистрибутивов Linux для встраиваемых систем на базе различных аппаратных архитектур.
* Описать проект, инструменты и возможности.
* Описать процесс сборки дистрибутива.
* Собрать и запустить дистрибутив Linux для x86.
* Рассмотреть процесс сборки с использованием web оболочки *Toaster*

# Описание проекта, инструментов и возможностей

*Yocto Project* – это совместный *Open Source*-проект разработки шаблонов, инструментов и методов для создания специальных дистрибутивов Linux для встраиваемых систем на базе различных аппаратных архитектур.

Будучи проектом совместного сотрудничества (иногда такие проекты называются "зонтичными"), *Yocto Project* охватывает различные составляющие процесса разработки. Эти составляющие именуются проектами в рамках общего проекта *Yocto Project* и включают в себя инструменты для сборки, метаданные инструкций по сборке (называемые рецептами), библиотеки, утилиты и графические интерфейсы.

Инструменты (Создание специальных дистрибутивов Linux для встраиваемых систем с помощью Yocto Project, 2013):

* *Poky* – это эталонная система сборки в рамках проекта *Yocto Project*. Она включает в себя *BitBake, OpenEmbedded-Core*, пакет поддержки платформы (*Board Support Package, BSP*), а также прочие пакеты и компоненты, объединенные в единую сборку. Название *Poky* также относится к эталонному дистрибутиву Linux, который создается этой системой сборки и может быть чрезвычайно минималистичным (*core-image-minimal*) или же представлять собой полноценную систему Linux с графической оболочкой (*core-image-sato*).
* Набор метаданных разделен на слои, каждый из которых обладает дополнительной функциональностью по отношению к нижележащим слоям. Базовый слой называется *OpenEmbedded*-Core (или *oe-core*) и содержит общие рецепты, классы и связанные с ними функции, необходимые для любой сборки. Эти сборки впоследствии можно настраивать под собственные нужды, добавляя новые слои поверх слоя *oe-core*.
* Пакет поддержки платформы содержит пакеты и драйверы, необходимые для создания Linux-дистрибутива для определенной платформы или архитектуры. Эти пакеты часто поддерживаются производителями компьютерного оборудования. BSP-пакеты являются интерфейсом между операционной системой Linux и аппаратной частью, на которой она запускается. Заметим, что можно также создавать BSP для виртуальных машин.
* *BitBake* – это система сборки. Она считывает рецепты (определенные наборы инструкций) и следует им – скачивает необходимые пакеты, компилирует их и создает результирующие загрузочные образы. *BitBake* совместно поддерживается проектами *Yocto Project* и *OpenEmbedded*.
* *Hob*. Для упрощения процесса разработки Linux для встраиваемых устройств в проекте *Yocto Project* было реализовано несколько различных методов, позволяющих работать с графической средой. Относительно новым дополнением к проекту является *Hob*. Это дополнение предоставляет в распоряжение разработчиков графический пользовательский интерфейс для *BitBake*, т. е. для процесса сборки. Оба этих компонента постоянно развиваются с учетом отзывов пользователей. На данный момент устарел.
* *Toaster* – веб оболочка для сборки дистрибутивов.
* *Embedded GLIBC (EGLIBC)* – это вариант библиотеки *GNU C Library (GLIBC)*, который был разработан для использования во встраиваемых системах. Особенностями *EGLIBC* являются меньший объем, настраиваемые компоненты и улучшенная поддержка кросс-компиляции и кросс-тестирования. *EGLIBC* входит в состав *Yocto Project*, но поддерживается своей собственной руководящей командой.
* Инструментарий для разработки приложений (*Application Development Toolkit, ADT*) позволяет разработчикам систем включать наборы SDK в создаваемые ими дистрибутивы с помощью инструментов *Yocto Project*. Впоследствии эти наборы *SDK* могут использовать сторонние разработчики для создания приложений для этих дистрибутивов. В состав *ADT* входят инструменты кросс-компиляции, утилиты для отладки и анализа производительности, а также сценарии эмуляции и поддержки *QEMU*. Кроме того, в *ADT* включен подключаемый модуль *Eclipse*, который могут использовать те, кто предпочитает работать в интегрированной среде разработки (*IDE*).
* *Autobuilder*: предназначен для автоматизации тестов и оценки качества продуктов, развиваемых на базе *Yocto Project*.
* *Cross-Prelink*: предназначен для предварительной компоновки в средах, использующих кросс-компиляцию, что позволяет повысить производительность программ.
* *Pseudo*: эмулирует доступ от имени пользователя root, что необходимо при создании конечного загрузочного образа.
* *Swabber*: определяет, не содержит ли сборка, выполненная с использованием кросс-компиляции, компоненты хостовой системы.
* *Build Appliance*: это виртуальная машина, в которой запущен *Hob*; позволяет получить представление о *Yocto Project* из первых рук тем, кто создает сборки на компьютерах под управлением операционной системы, отличной от Linux (Примечание: на сегодняшний день инструменты сборки *Yocto Project* можно использовать только в ОС Linux).

# Описание процесса сборки дистрибутива

Для работы с системой сборки *poky*, сначала нужно ее установить на компьютер.

Существует 2 способа получения *poky*:

1. Загрузка *tar* файла с последней протестированной версией *Yocto Project* со страницы загрузки проекта

$ wget \

http://downloads.yoctoproject.org/releases/yocto/yocto-1.2/poky-denzil-7.0.tar.bz2

$ tar xjf poky-denzil-7.0.tar.bz2

$ cd poky-denzil-7.0

1. Получение последней версии (или любой отдельной ветки) при помощи *git*, хотя главная ветка разработки может оказаться менее стабильной по сравнению с протестированной версией, содержащейся в *tar* файле.

$ git clone git://git.yoctoproject.org/poky.git

$ cd poky

Инициализация рабочего окружения:

Сначала нужно инсталлировать все необходимые пакеты из репозитория программного обеспечения (Yocto Project Mega-Manual, 2015).

$ sudo apt-get install sed wget subversion git-core coreutils \

unzip texi2html texinfo libsdl1.2-dev docbook-utils fop gawk \

python-pysqlite2 diffstat make gcc build-essential xsltproc \

g++ desktop-file-utils chrpath libgl1-mesa-dev libglu1-mesa-dev \

autoconf automake groff libtool xterm libxml-parser-perl

Задание значений переменных окружения командной оболочки с помощью входящего в состав поставки сценария:

$ cd poky

$ . ./oe-init-build-env

Создание образа:

После задания значений переменных окружения рабочей директорией станет */build*

### Shell environment set up for builds. ###

You can now run 'bitbake <target>'

Common targets are:

core-image-minimal

core-image-sato

meta-toolchain

adt-installer

meta-ide-support

You can also run generated qemu images with a command like 'runqemu qemux86'

Далее задаются настройки сборки в файле *conf/local.conf*

По умолчанию введены следующие настройки:

MACHINE ??= "qemux86"

DISTRO ?= "poky"

PACKAGE\_CLASSES ?= "package\_rpm"

EXTRA\_IMAGE\_FEATURES = "debug-tweaks"

USER\_CLASSES ?= "buildstats image-mklibs"

PATCHRESOLVE = "noop"

BB\_DISKMON\_DIRS = "\

STOPTASKS,${TMPDIR},1G,100K \

STOPTASKS,${DL\_DIR},1G,100K \

STOPTASKS,${SSTATE\_DIR},1G,100K \

STOPTASKS,/tmp,100M,100K \

ABORT,${TMPDIR},100M,1K \

ABORT,${DL\_DIR},100M,1K \

ABORT,${SSTATE\_DIR},100M,1K \

ABORT,/tmp,10M,1K"

PACKAGECONFIG\_append\_pn-qemu-native = " sdl"

PACKAGECONFIG\_append\_pn-nativesdk-qemu = " sdl"

ASSUME\_PROVIDED += "libsdl-native"

CONF\_VERSION = "1"

Для оптимизации сборки можно задать следующие параметры, связанные с параллельной обработкой, для ускорения процесса сборки. На этом шаге зададим значения этих параметров, равными двукратному числу ядер процессора (например, 8 для 4-ядерного процессора). Для данной работы был использован *VPS (Virtual Private Server*) сервер *DigitalOcean.com*, с характеристиками – 12 CPU, 32 GB RAM, поскольку сборка занимает много времени.

BB\_NUMBER\_THREADS = "24"

PARALLEL\_MAKE = "-j 24"

Для старта сборки используется команда *bitbake*. Это сборка с *GUI SATO*:

bitbake core-image-sato

После этого запустится сборка дистрибутива.

Запуск виртуальной машины QEMU с собранным образом осуществляется следующей командой:

runqemu qemux86

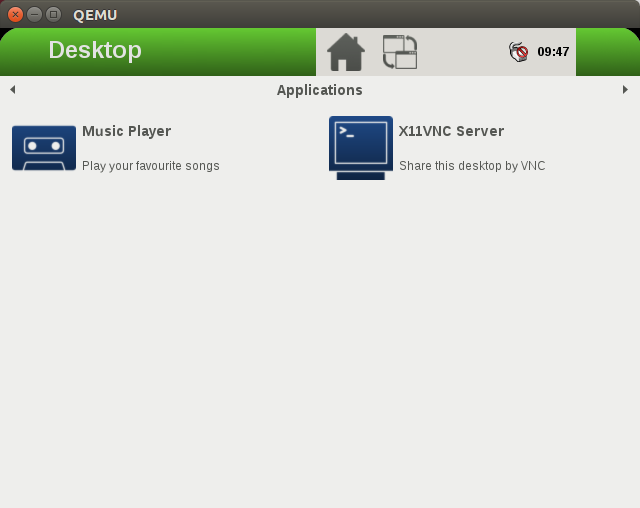


Рисунок 1. QEMU после запуска собранного образа core-image-sato

# Процесс сборки с использованием web оболочки Toaster

Для запуска веб-интерфейса *Toaster* необходимо сделать следующее (Toaster User Manual, 2015):

Установить пакетный менеджер *python - pip*:

sudo apt-get install python-pip

Далее установить необходимые зависимости:

pip install –r /home/user/poky/bitbake/toaster-requirements.txt

И запустить *Toaster*:

. /home/user/poky/bitbake/bin/toaster

Далее запустится web-интерфейс.

Создадим новый проект:

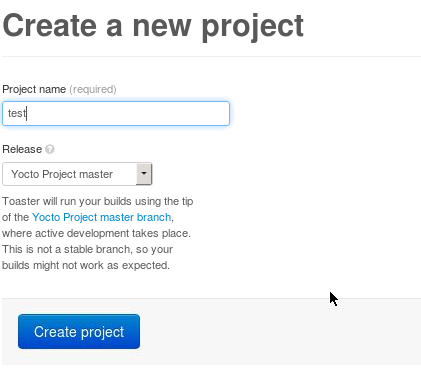


Рисунок 2. Toaster. Создание нового проекта

По умолчанию заданы следующие настройки:

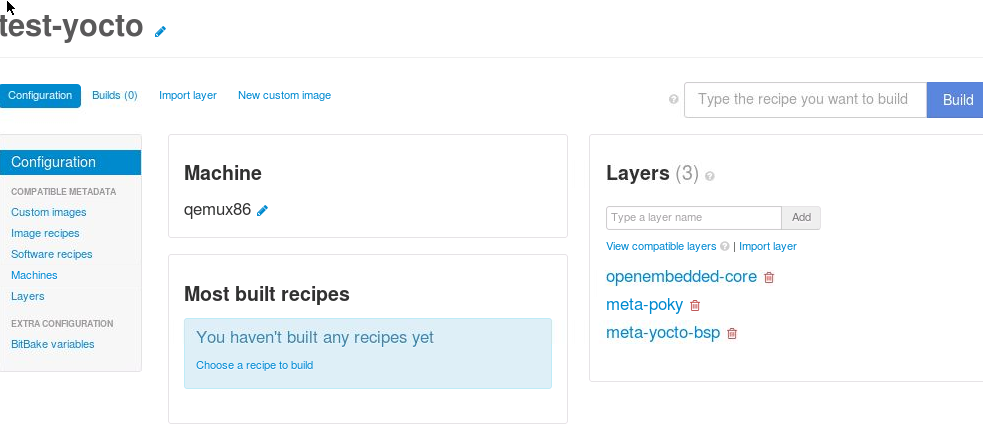


Рисунок 3. Toaster. Начальные натройки проекта

Для того чтобы добавить в дистрибутив нужный пакет нужно его найти в доступных слоях. Например, добавим *apache2*. Выбираем слой *meta-webservers*. Далее следует нажать *Add build*

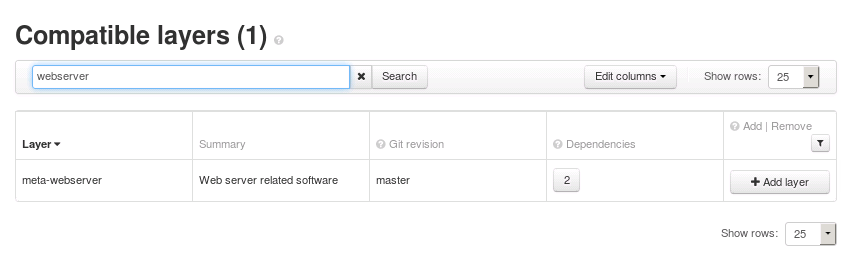


Рисунок 4. Toaster. Доступные слои.

После этого нужно согласиться установить зависимости:

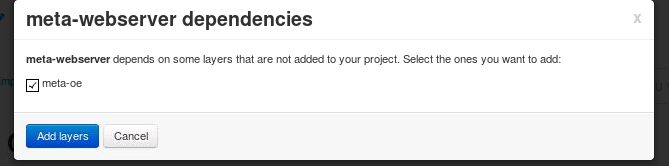


Рисунок 5. Toaster. Диалоговое окно. Предложение установить зависимости

Далее соберем *apache2* рецепт:

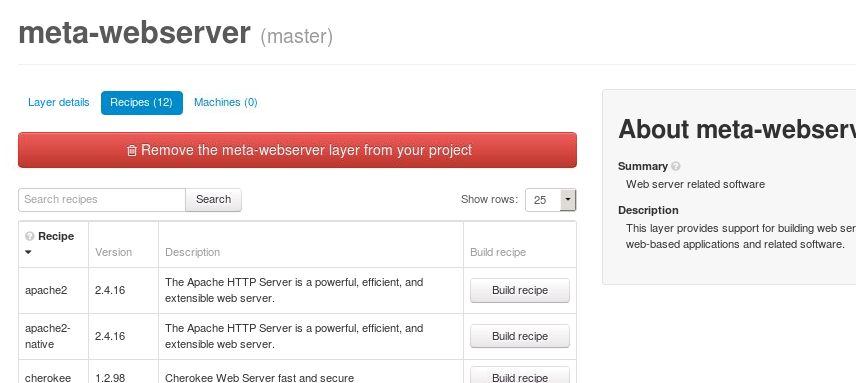


Рисунок 6. Toaster. Сборка рецепта слоя meta-webserver

Для того чтобы образ, который будет собран включал *apache2* нужно задать следующие параметры:

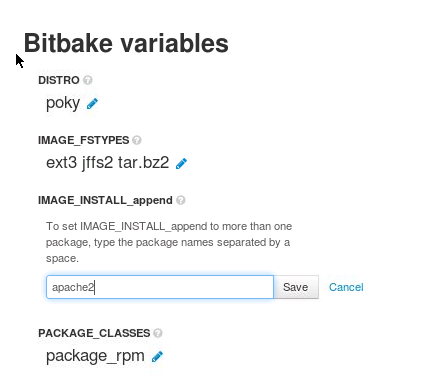


Рисунок 7. Toaster. Настройки BitBake

После этого можно собирать образ, который будет включать *apache2*.

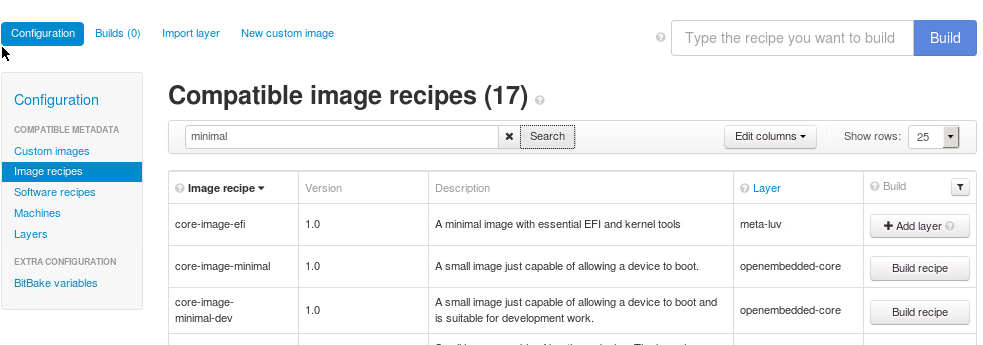


Рисунок 8. Toaster. Сборка образа

Собранный образ запускается, как и в предыдущем примере:

runqemu qemux86

# Вывод

*Yocto Project* – это совместный *Open Source*-проект разработки шаблонов, инструментов и методов для создания специальных дистрибутивов Linux для встраиваемых систем на базе различных аппаратных архитектур.

Будучи проектом совместного сотрудничества (иногда такие проекты называются "зонтичными"), *Yocto Project* охватывает различные составляющие процесса разработки.

Таким образом *yocto* позволяет гибко создавать дистрибутивы для различных архитектур. Множество документации *yocto* оправдывает высокий порог вхождения.

# Список литературы

*Toaster User Manual*. (апрель 2015 г.). Получено из www.yoctoproject.org: http://www.yoctoproject.org/docs/2.0.1/toaster-manual/toaster-manual.html

*Yocto Project Mega-Manual*. (апрель 2015 г.). Получено из www.yoctoproject.org: http://www.yoctoproject.org/docs/2.0.1/mega-manual/mega-manual.html

*Создание специальных дистрибутивов Linux для встраиваемых систем с помощью Yocto Project*. (16 январь 2013 г.). Получено из http://www.ibm.com/: http://www.ibm.com/developerworks/ru/library/l-yocto-linux/

# Список литературы

1. http://www.yoctoproject.org/docs/current/yocto-project-qs/yocto-project-qs.html

2. http://rus-linux.net/MyLDP/BOOKS/Architecture-Open-Source-Applications/Vol-2/yocto-01.html

3. http://www.ibm.com/developerworks/ru/library/l-yocto-linux/