Developing applications on Yocto

Lianhao Lu
Intel Corporation
Feb. 29th, 2012
Agenda

● Embedded Linux Development

● The Yocto Project Offerings For Embedded Linux Development

● The Yocto Project Eclipse Plug-in
  For System Developer
  For Application Developer
Embedded Linux Development Flow

System Developer

Base platform

Profile

Toolchain

Sysroot

Rootfs

Kernel

Application Developer

App project

Apps

Packages

Customized platform

Runtime image

System Developer
Embedded Linux Development

● System developers:
  • Develop the Linux system for the targeted embedded device
  • Tasks:
    ✔ Build profile customization through package selection
    ✔ Image footprint tuning
    ✔ Reproducible build with flexibility on customization, e.g. target architecture, package format, etc
    ✔ Build toolchain for application developers

● Application developers:
  • Develop applications running on the targeted embedded device
  • Tasks:
    ✔ Use cross toolchain
    ✔ Take advantage of sysroot setup
    ✔ Remote debug application on target (real HW or emulator)
    ✔ Performance tuning through profiling/tracing tools

A framework that streamline the development flow is highly desirable
The Yocto Project Offerings for Embedded Linux Development

The Yocto Project

- BitBake + meta-data
- ADT
- Eclipse plug-in
- ...

It's not an embedded Linux distribution – it creates a custom one for you
The Yocto Project Offerings for Embedded Linux Development

● The Build system and meta-data

  ▪ BitBake - widely adopted build system by the embedded Linux developers

  ▪ Meta-data contains recipe and configuration files

  ▪ Easily customization/extension of the core meta-data through layers

  ▪ HOB – A graphical user interface for BitBake

You don't need to be an expert of BitBake to be able to customize your build and image
The Yocto Project Offerings for Embedded Linux Development (cont.)

● The Application Development Kit (ADT)

- Cross toolchain for the target device
  - ✔ Support sysroot setup
  - ✔ Optimized for autotool based projects

- Qemu emulator
  - ✔ Can be booted through unfs
  - ✔ Rootfs is extracted as sysroot

- Tools for target analysis, profiling and tracing

- ADT-installer
Prepare the development environment

● Where to start:

● Installing ADT
  ● Using ADT installer
  ● Using cross-toolchain tarball
  ● Using Yocto build tree
The Yocto Project Eclipse Plug-in

● An IDE environment to streamline the development flow
  ✔ Wizard
  ✔ Template

● Based on open source solutions
  ✔ Eclipse communities' CDT, RSE, TCF and LinuxTools projects
  ✔ BitBake Commander Project

● Within one IDE, users can fully benefit from Yocto Project offerings:
  ✔ BitBake (through Hob)
  ✔ Meta-data
  ✔ ADT
  ✔ Qemu
  ✔ Tracing and profiling tools
Installing Eclipse

- Install latest Eclipse framework along with the following feature/plugins:
  - C/C++ Development Tools (CDT) and C/C++ Remote Launch
  - Autotools support for CDT and LTTng
  - Remote System Explorer (RSE), Target Management Terminal
  - Target Communication Framework (TCF)

- Make sure tcf-agent is running on the target system

- Install Yocto Eclipse Plugin
  - From Update Site
  - From latest source
The Yocto Project Eclipse Plug-in For System Developer

Without plug-in:
Everything has to be done from command line

Step 1: Git clone Yocto Project meta-data
Step 2: Edit recipe files using your prefer editor: emacs, vim, etc
Step 3: Source oe-init-build-env to setup your build directory
Step 4: Edit conf/local.conf file to configure the Yocto build
Step 5: Run bitbake command to kick off the build
Or
Step 4: Or use Hob to facilitate further build customization and run Yocto build
The Yocto Project Eclipse Plug-in For System Developer

Step 1: Create Yocto BitBake Commander Project for Yocto Project metadata (Note: Collaborate with other plug-ins, e.g. egit)
Step 2: Customize meta-data recipe files.

✔ Navigate the meta-data in the project tree view
✔ "Yocto BitBake Recipe Editor" with keywords highlighted
✔ New Recipe Wizard allows quickly create new recipe files
The Yocto Project Eclipse Plug-in For System Developer

Step 3: Launch Hob
✔ Setup a separate build area for the customized meta-data
✔ Through Hob further customize and configure your build and image output
✔ Run the build from Hob
The Yocto Project Eclipse Plug-in For System Developer

Hob

Select a machine
This is the profile of the target machine for which you are building the image.

Select a base image
Base images are a starting point for the type of image you want. You can build them as they are or customize them to your specific needs.

View Recipes
Add/remove recipes and collections

View Packages
Add/remove packages

Build Packages or Just bake
The Yocto Project Eclipse Plug-in For System Developer

Demo
The Yocto Project Eclipse Plug-in For Application Developer

Without plug-in:

Step 1: setup your cross toolchain and sysroot for cross development
Step 2: Create your Makefile or autotool based project
  ● Best with autotool based project, just pass host options to configure, e.g. ./configure
    host=i586-poky-linux --with-libtool-sysroot=/home/lulianhao/rootfs/x86
  ● For other projects should ensure the cross tools are used, e.g. CC=i586-poky-linux-gcc and
    LD=i586-poky-linux-ld in makefile
Step 3: Compile your project
Step 4 (optional): Bring up Qemu emulator through command line
Step 5: Deploy your application to remote target: rcp, scp, rsync, etc.
Step 6: Setup cross debugging against desired target, Qemu or real HW
  ● Start gdbserver on target
  ● Run cross-gdb on host side to connect to remote target
Step 7: Some target analysis task, tracing, profiling, etc.
  ● Follow each tool's special setup for remote launch or interaction from host

A very complex task if doing everything on your own. Can dramatically slow down development cycle.
The Yocto Project Eclipse Plug-in For Application Developer

Step 1: Setup your cross toolchain and sysroot for cross development
Step 2: Setup your Eclipse IDE with Yocto Project plug-in installed
Step 3: Configure Yocto Project ADT plug-in for IDE
Step 4: Pick one of ADT autotoool based project templates
Step 5: Change project cross develop settings if needed
Step 6: Work on your project, configure and compile use cross dev settings
Step 7: Use auto-created qemu launcher for the target to launch qemu
Step 8: Finish the auto-created remote debug configuration template for the project.
The Yocto Project Eclipse Plug-in For Application Developer

Step 9: Launch the remote debug session
The Yocto Project Eclipse Plug-in For Application Developer

Step 10: Use the tools through YoctoTools. The tools suite contains the following essential tools that provide target analytical capabilities:

- PowerTop
- LatencyTop
- Oprofile
- Perf
- Lttng-ust
- SystemTap
The Yocto Project Eclipse Plug-in For Application Developer

Demo
Information

Yocto Project:
http://www.yoctoproject.org/
http://www.yoctoproject.org/documentation
http://www.yoctoproject.org/community

ADT manual

Yocto Project Eclipse plug-in video
http://vimeo.com/30557368

ELC 2011 ADT Video
The Yocto project and its application development toolkit (ADT) - The answer to effective embedded application development
INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. EXCEPT AS PROVIDED IN INTEL’S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO SALE AND/OR USE OF INTEL PRODUCTS, INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT.

Intel may make changes to specifications, product descriptions, and plans at any time, without notice.

All dates provided are subject to change without notice.

Intel is a trademark of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2012, Intel Corporation. All rights are protected.