PTXdist Application Note

Building OSELAS-Crosstool Toolchains

Before we can start building our first userland we need a cross toolchain. On Linux, toolchains are no monolithic beasts. Most parts of what we need to cross compile code for the embedded target comes from the *GNU Compiler Collection*, `gcc`.

This application note describes how to build OSELAS.Crrosstool toolchains to be used for projects handled by PTXdist

Building Toolchains

Before we can start building our first userland we need a cross toolchain. On Linux, toolchains are no monolithic beasts. Most parts of what we need to cross compile code for the embedded target comes from the *GNU Compiler Collection*, `gcc`. The gcc packet includes the compiler frontend, `gcc`, plus several backend tools (cc1, g++, ld etc.) which actually perform the different stages of the compile process. `gcc` does not contain the assembler, so we also need the *GNU Binutils package* which provides lowlevel stuff.

Cross compilers and tools are usually named like the corresponding host tool, but with a prefix – the *GNU target*. For example, the cross compilers for ARM and powerpc may look like

- `arm-softfloat-linux-gnu-gcc`
- `powerpc-unknown-linux-gnu-gcc`

With these compiler frontends we can convert e.g. a C program into binary code for specific machines. So for example if a C program is to be compiled natively, it works like this:

```
~$ gcc test.c -o test
```

To build the same binary for the ARM architecture we have to use the cross compiler instead of the native one:

```
~$ arm-softfloat-linux-gnu-gcc test.c -o test
```

Also part of what we consider to be the "toolchain" is the runtime library (libc, dynamic linker). All programs running on the embedded system are linked against the libc, which also offers the interface from user space functions to the kernel.

The compiler and libc are very tightly coupled components: the second stage compiler, which is used to build normal user space code, is being built against the libc itself. For example, if the target does not contain a hardware floating point unit, but the toolchain generates floating point code, it will fail. This is also the case when the toolchain builds code for i686 CPUs, whereas the target is i586.

So in order to make things working consistently it is necessary that the runtime libc is identical with the libc the compiler was built against.

PTXdist doesn't contain a pre-built binary toolchain. Remember that it's not a distribution but a development tool. But it can be used to build a toolchain for our target. Building the toolchain usually has only to be done once. It may be a good idea to do that over night, because it may take several hours, depending on the target architecture and development host power.

Using Existing Toolchains

If a toolchain is already installed which is known to be working, the toolchain building step with PTXdist may be omitted.

> ⚠ The generic board support packages shipped for PTXdist have been tested with the OSELAS.Crosstool toolchains built with the same PTXdist version. So if an external toolchain is being used which isn't known to be stable, a target may fail. Note that not all compiler versions and combinations work properly in a cross environment.

Toolchain Building

PTXdist-0.10.6 handles toolchain building as a simple project, like all other projects, too. If we already installed the `ptxdist-0.10.6-projects.tgz` archive while installing PTXdist-0.10.6 we now can clone one of the crosstool toolchain projects to build the required toolchain for the other generic projects.

A PTXdist project generally allows to build into some project defined directory; all OSELAS.Toolchain projects that come with PTXdist are configured to use the standard installation paths mentioned above.

All OSELAS.Toolchain projects install their result into `/opt/ptxdist-0.10.6/`.

> **(!)** Usually the `/opt` directory is not world writable. So in order to build our OSE-LAS.Crosstool toolchain into that directory we need to use a root account to change the permissions so that the user can write into
> ```
> mkdir /opt/ptxdist-0.10.6
> chown <username> /opt/ptxdist-0.10.6
> chmod a+rwx /opt/ptxdist-0.10.6
> ```

Building the OSELAS.Crosstool toolchain

To compile and install a crosstool toolchain we have to clone the crosstool project first. To get a list of predefined projects to clone, run this command:

```
˜$ ptxdist projects
-------------------- Available PTXdist Projects:  ----------------------
OSELAS.BSP-Pengutronix-GenericArmGlibc-2
OSELAS.BSP-Pengutronix-GenericI586Glibc-2
OSELAS.BSP-Pengutronix-GenericPowerpcGlibc-2
OSELAS.Crosstool-1.0.1
------------------------------------------------------------------------
```

The `OSELAS.Crosstool-1.0.1` is the project in question for us.

```
˜$ ptxdist clone projects OSELAS.Crosstool-1.0.1 toolchain
```

This will clone the `OSELAS.Crosstool-1.0.1` project into the local `toolchain` folder. In the next step we must choose target's architecure to build the correct cross compiler. To do so, we change into the new directory and run the `ls` command to see, what toolchains are provided:

```
˜$ cd toolchain
˜/toolchain$ ls toolchain*
toolchain_arm-softfloat-linux-gnu-4.0.2_glibc_2.3.6_linux_2.6.14.ptxconfig
toolchain_i586-unknown-linux-gnu-4.0.2_glibc_2.3.6_linux_2.6.17.ptxconfig
toolchain_powerpc-unknown-linux-gnu-4.0.2_glibc-2.3.6_linux_2.6.13.ptxconfig
```

In this application note we use the ARM architecture as our example to build. Selecting will be done by the *select* command to PTXdist.

```
˜/toolchain$ ptxdist select
    toolchain_arm-softfloat-linux-gnu-4.0.2_glibc_2.3.6_linux_2.6.14.ptxconfig
```

That's all. Now we are going to build this toolchain by entering:

```
˜/toolchain$ ptxdist go
```

When PTXdist starts to build the toolchain we have to go to our boss and tell him that it's probably time to go home for the day. Even on reasonably fast machines the time to build an OSELAS.Crosstool-1.0.1 toolchain is something like around 45 minutes up to a few hours.

Measured times on different machines:

- Single Pentium 2.5GHz, 2 GiB RAM: about 2 hours

- Dual Athlon 2.1GHz, 2 GiB RAM: about 1 hour 20 minutes

- Dual Quad-Core-Pentium 1.8GHz, 8 GiB RAM: about 25 minutes

When the OSELAS.Crosstool-1.0.1 project build is finished, PTXdist is ready for prime time and we can continue with our first project.

Freezing the Toolchain

As we built and install this toolchain with regular user rights we should modify the permissions as a last step to avoid any later manipulation. To do so we could set all toolchain files to read only or changing recursivley the owner of the whole installation to user root.

This is an important step for reliability. Do not omit it!

Building additional Toolchains

The OSELAS.Crosstool-1.0.1 bundle comes with various predefined toolchains. To build additional toolchains we only have to clean our current toolchain projekt, removing the current `ptxconfig` link and creating a new one.

```
~/toolchain$ ptxdist clean
~/toolchain$ rm ptxconfig
~/toolchain$ ptxdist select
       toolchain_any_another_toolchain.ptxconfig
~/toolchain$ ptxdist go
```

All toolchains will be installed side by side architecture dependend into directory

$$/opt/ptxdist-0.10.6/architecture\_part.$$

Additional questions?

Below a list of locations where you can get help in case of trouble or questions how to do something special within PTXdist or general questions about Linux in the embedded world.

Mailing Lists

About PTXdist in special

This is an english language public mailing list for questions about PTXdist. See web site

<div align="center">

http://www.pengutronix.de/mailinglists/index_en.html

</div>

how to subscribe to this list. If you want to search through the mailing list archive, visit

<div align="center">

http://www.mail-archive.com/

</div>

and search for the list *ptxdist*.

About embedded Linux in general

This is a german language public mailing list for general questions about Linux in embedded environments. See web site

<div align="center">

http://www.pengutronix.de/mailinglists/index_de.html

</div>

how to subscribe to this list. Note: You also can send english language mails.

News Groups

About Linux in embedded environments

This is an english language news group for general questions about Linux in embedded environments.

<div align="center">

comp.os.linux.embedded

</div>

About general Unix/Linux questions

This is a german language news group for general questions about Unix/Linux programming.

<div align="center">

de.comp.os.unix.programming

</div>

Chat/IRC

About PTXdist in special

<div align="center">

irc.freenode.net:6667

</div>

Create a connection to the **irc.freenode.net:6667** server and enter the chat group **#ptxdist**. This is an english language group to answer questions about PTXdist. Best time to meet somebody in there is at europeen daytime.

Miscellaneous

Online Linux Kernel Cross Reference

A powerful cross reference to be used online.

<center>http://lxr.linux.no/blurb.html</center>

U-Boot manual (partially)

Manual how to survive in an embedded environment and how to use the U-Boot on target's side

<center>http://www.denx.de/wiki/DULG</center>

Commercial Support

You can order immediate support through customer specific mailing lists, by telephone or also on site. Ask our sales representative for a price quotation for your special requirements.

Contact us at:

<center>

Pengutronix
Hannoversche Strasse 2
D-31134 Hildesheim
Germany
Phone: +49 - 51 21 / 20 69 17 - 0
Fax: +49 - 51 21 / 20 69 17 - 9

</center>

or by electronic mail:

<center>**sales@pengutronix.de**</center>

Do you want to help improving this document?
Send your suggestions to *jbe@pengutronix.de*