PTXdist Application Note

Building OSELAS.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.Toolchains() to be used for projects handled by PTXdist

Toolchains

Abstract

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 OSELAS.BoardSupport() Packages shipped for PTXdist have been tested with the OSELAS.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.

Every OSELAS.BoardSupport() Package checks for its OSELAS.Toolchain it's tested against, so using a different toolchain vendor requires an additional step:

Open the OSELAS.BoardSupport() Package menu with:

`˜$ ptxdist menuconfig`

and navigate to `PTXdist Config`, `Architecture` and `Check for specific toolchain vendor`. Clear this entry to disable the toolchain vendor check.

Building a Toolchain

PTXdist-1.0.2 handles toolchain building as a simple project, like all other projects, too. So we can download the OSELAS.Toolchain bundle and build the required toolchain for the OSELAS.BoardSupport() Package.

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 below.

All OSELAS.Toolchain projects install their result into `/opt/OSELAS.Toolchain-1.1.1/`.

!   Usually the `/opt` directory is not world writable. So in order to build our OSELAS.Toolchain into that directory we need to use a root account to change the permissions so that the user can write (`mkdir /opt/OSELAS.Toolchain-1.1.1 ; chown <username> /opt/OSELAS.Toolchain-1.1.1; chmod a+rwx /opt/OSELAS.Toolchain-1.1.1`).

Building a sample OSELAS.Toolchain

To compile and install an OSELAS.Toolchain we have to extract the OSELAS.Toolchain archive, change into the new folder, configure the compiler in question and start the build.

As an example we will build the

i586-unknown-linux-gnu_gcc-4.1.2_glibc-2.5_linux-2.6.18.

toolchain.

The steps to do so are:

```
~$ tar xf OSELAS.Toolchain-1.1.1.tar.bz2
~$ cd OSELAS.Toolchain-1.1.1
~/OSELAS.Toolchain-1.1.1$ ptxdist select
  ptxconfigs/i586-unknown-linux-gnu_gcc-4.1.2_glibc-2.5_linux-2.6.18.ptxconfig
~/OSELAS.Toolchain-1.1.1$ ptxdist go
```

At this stage 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.Toolchain is something like around 30 minutes up to a few hours.

Measured times on different machines:

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

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

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

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

Freezing the Toolchain

As we build 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.Toolchain-1.1.1 bundle comes with various predefined toolchains. Refer the `ptxconfigs/` folder for other definitions. To build additional toolchains we only have to clean our current toolchain projekt, removing the current `ptxconfig` link and creating a new one.

```
~/OSELAS.Toolchain-1.1.1$ ptxdist clean
~/OSELAS.Toolchain-1.1.1$ rm ptxconfig
~/OSELAS.Toolchain-1.1.1$ ptxdist select
        ptxconfigs/any_another_toolchain_def.ptxconfig
~/OSELAS.Toolchain-1.1.1$ ptxdist go
```

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

```
/opt/OSELAS.Toolchain-1.1.1/architecture_part.
```

Different toolchains for the same architecture will be installed side by side version dependend into directory

```
/opt/OSELAS.Toolchain-1.1.1/architecture_part/version_part.
```

Additional questions?

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

Mailing Lists

About PTXdist in particular

This is an English language public mailing list for questions about PTXdist. See

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

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

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

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

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

how to subscribe to this list. Note: You also can send mails in English.

News Groups

About Linux in embedded environments

This is an English newsgroup 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 newsgroup for general questions about Unix/Linux programming.

<div align="center">

de.comp.os.unix.programming

</div>

Chat/IRC

About PTXdist in particular

<div align="center">

irc.freenode.net:6667

</div>

Create a connection to the **irc.freenode.net:6667** server and enter the chatroom **#ptxdist**. This is an English room to answer questions about PTXdist. Best time to meet somebody there is at European daytime.

Miscellaneous

Online Linux Kernel Cross Reference

A powerful online cross reference.

<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>

This is a Pengutronix Application Note

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