

Creating PS3 toolchains on Gentoo

Contributed by Administrator
Friday, 15 December 2006
Last Updated Saturday, 31 March 2007

So, now you have a Gentoo install on a PS3, the next big task is to install the development tools. At the moment the process for creating the toolchains is in a state of flux, I'll try to keep this up to date but keep an eye out for some "official" documentation appearing at some point.

The PPE toolchain

For development on the PS3, you need tool chains for both the PPE and the SPE's of the Cell processor. You already have a working PPE toolchain, as that's what you used to build your gentoo install, so all you need is the SPE toolchain. The SPE toolchain

The simplest way to do this is to go to the Barcelona Super Computer Center and download binaries, well this is Gentoo, we never do things the simple way if we have the option of spending all day compiling! Installing the lu_zero overlay

Building the SPE toolchain is actually quite simple as the developer Lu_zero has created an overlay containing everything we need. If you haven't installed his overlay already, there are instructions here:

http://www.daniel.jp/joomla/info/ps3/installing-the-lu_zero-overlay.html Installing the toolchain overlay

When I originally wrote this tutorial, the needed gcc and binutils were in Lu_zero's overlay, currently you can use the masked binutils in portage, and the gcc-4.3 in the toolchains overlay at:

<http://overlays.gentoo.org/proj/toolchain/timeline>

To install the overlay use the tool layman, there are user docs at:

<http://www.gentoo.org/proj/en/overlays/userguide.xml>

Updating your PPE toolchain

Before we create the SPE toolchain, we can take the chance to update our existing PPE toolchain, right up to the bleeding edge:)

First compile the crossdev utility: `# emerge crossdev`

Edit your `/etc/portage/package.keywords` and add: `sys-devel/gcc -*`
`sys-libs/newlib`
`sys-devel/binutils -*`
`sys-devel/patch`

To use the gcc-4.3 in the overlay you will need to unmask it, so add to `/etc/portage/package.unmask` `sys-devel/gcc`

One little thing to watch out for... The GCC you are about to compile is beta so there is a hoop you need to jump through: `export I_PROMISE_TO_SUPPLY_PATCHES_WITH_BUGS=1`

Don't forget to do that again before running crossdev spu if it is in a new session!!

Compile them: `# emerge binutils`
`# emerge gcc`

And then switch over to using the new compiler: `# gcc-config -l`

- [1] powerpc64-unknown-linux-gnu-3.4.6
- [2] powerpc64-unknown-linux-gnu-3.4.6-hardened
- [3] powerpc64-unknown-linux-gnu-3.4.6-hardenednopie
- [4] powerpc64-unknown-linux-gnu-3.4.6-hardenednopiessp
- [5] powerpc64-unknown-linux-gnu-3.4.6-hardenednossp
- [6] powerpc64-unknown-linux-gnu-4.1.1
- [7] powerpc64-unknown-linux-gnu-4.3.0-alpha20061111 *

`# gcc-config 7`

As of 19-12-06 there is a little hack needed to install the spu g++, we need to edit toolchain.eclass and add a change by lu_zero that hasn't yet been committed.

Open /usr/portage/eclass/toolchain.eclass, look through it for:

```

        export gcc_cv_libc_provides_ssp=yes
        configcc="${configcc} --disable-libssp"
    fi

fi

# GTK+ is preferred over xlib in 3.4.x (xlib is unmaintained
# right now). Much thanks to <csn@gnu.org> for the heads up.
# Travis Tilley <lv@gentoo.org> (11 Jul 2004)

```

And add a new if block like so:

```

        export gcc_cv_libc_provides_ssp=yes
        configcc="${configcc} --disable-libssp"
    fi

```

```

fi

if [[ ${CTARGET} == *-elf ]] ; then
    configcc="${configcc} --with-newlib"
fi

```

```

# GTK+ is preferred over xlib in 3.4.x (xlib is unmaintained
# right now). Much thanks to <csn@gnu.org> for the heads up.
# Travis Tilley <lv@gentoo.org> (11 Jul 2004)

```

And then you can compile the SPE toolchain using the crossdev utility.

The crossdev command we need to use is a little more complicated than usual, this is to make sure we use the right version of newlib and gcc and override the defaults in crossdev :# CFLAGS=-O2 crossdev --l 1.15.0 --g 4.3.0_alpha20061216 spu Installing libspe2

The lu_zero overlay contains both libspe and libspe2, and so to install them first edit your /etc/portage/package.keywords and add:sys-libs/libspe sys-libs/libspe2

Before building them or the samples package, there are a couple of things you need to take care of.

1) Make sure your compiler is set to gcc-4.1.1.

2) Get hold of the script embedspu.sh make it executable and save it as /usr/bin/embedspu.# cp embedspu.sh /usr/bin/embedspu
chmod +x /usr/bin/embedspu

And then install libspe and libspe2 with the command:# emerge libspe libspe2

Finally you need make sure elfspe is registered to binfmt:# rc-update add elfspe default
/etc/init.d/elfspe start

So you should now be ready to try some Cell development!Testing the toolchains

We have one more little bit of house keeping to take care of - spufs.

Edit your /etc/fstab and add the following line:none /spu spufs defaults 0 0

and then edit /etc/modules.autoload.d/kernel-2.6 and add the module:# /etc/modules.autoload.d/kernel-2.6: kernel modules to load when system boots.

```

#
# Note that this file is for 2.6 kernels.
#
# Add the names of modules that you'd like to load when the system

```

starts into this file, one per line. Comments begin with # and # are ignored. Read man modules.autoload for additional details.
For example:
aic7xxx
spufs

then make the mount point:# mkdir /spu

and then load the module:# modprobe spufs

OK, Helloworld time!

If you go to www.cellperformance.com there is a helloworld app.

Download the ppu and spu code to a convenient location and then compile them:# gcc ppu_hello.c -lspe2 -o ppu_hello
spu-elf-gcc spu_hello.c -o spu_hello

And then with luck:# ./ppu_hello
Hello, World! (From SPU:0)

And you should also test elfspe2:# elfspe2 spu_hello
Hello, World! (From SPU:0)

If that doesn't work, the first thing to check is that spufs is mounted. Try: # mount | grep spu
none on /spu type spufs (rw)

If you don't get that output examine your fstab, and:# mount -t spufs none /spu

:)

Also, you can install the SCE SPE example applications from the lu_zero overlay.

Before you can do so, ensure that /etc/init.d/elfspe has been started.

Edit your /etc/portage/package.keywords and add: app-misc/spe-samples

And install them with the command: # emerge spe-samples