

Cross-Compile guide for MySQL

The first step to cross-compiling MySQL is setting up your OpenEmbedded environment. Before actually installing OpenEmbedded, you will need various packages. Note that you need Python 2, not 3.

```
> sudo apt-get install sed wget cvs subversion git-core \  
  coreutils unzip texi2html texinfo docbook-utils \  
  gawk python-pysqlite2 diffstat help2man make gcc build-essential g++ \  
  desktop-file-utils chrpath  
> sudo apt-get install libxml2-utils xmlto python-psyco apr
```

Optionally, you can install the following for extra documentation.

```
> sudo apt-get install docbook
```

However, requirements may change over time. You can consult the following page for up to date information. http://www.openembedded.org/wiki/Required_software. The rest of this guide will refer to the base directory where you install OpenEmbedded as `/stuff/`. This directory needs to be a location that has no symbolic links above it. We now setup the build directory.

```
> mkdir -p /stuff/build/conf  
> cd /stuff/
```

Next, download and place BitBake in your installation directory.

```
> wget http://download.berlios.de/bitbake/bitbake-1.10.2.tar.gz  
> tar -xzvf bitbake-1.10.2.tar.gz  
> mv bitbake-1.10.2 bitbake
```

After extracting BitBake, retrieve OpenEmbedded from git.

```
> git clone git://github.com/openembedded/openembedded.git
```

Now that we have all the necessary files, we create the local configuration file for OpenEmbedded. You can edit `local.conf` with your favourite text editor as well.

```
> cd /stuff/  
> cp openembedded/conf/local.conf.sample build/conf/local.conf  
> vi build/conf/local.conf
```

```
BBFILES := "/stuff/openembedded/recipes/*/*.bb"  
BBMASK = "/(nonworking|obsolete)/"  
MACHINE = "embedsky"  
DISTRO = "angstrom-2010.x"  
BB_NUMBER_THREADS = "2" # Can be set to more threads if available
```

Another distribution can be used if you're having trouble building recipes, but Angstrom worked well for us. Remember to remove the following line from the bottom of your file or BitBake will not run.

```
REMOVE_THIS_LINE:="$#{@bb.fatal('Read the comments in your conf/local.conf')}}"
```

We will also create our own machine configuration for our board, but you can probably use any provided board configuration file that has the appropriate architecture. In our case, we're looking for a tune-arm920 include file.

```
> vi /stuff/openembedded/conf/machine/embedsky.conf
```

```
TARGET_ARCH = "arm"
require conf/machine/include/tune-arm920t.inc
ROOT_FLASH_SIZE = "256"
VOLATILE_STORAGE_SIZE = "64"
MACHINE_GUI_CLASS = "smallscreen"
MACHINE_DISPLAY_WIDTH_PIXELS = "240"
MACHINE_DISPLAY_HEIGHT_PIXELS = "320"
MACHINE_FEATURES = "kernel26 touchscreen apm alsa usb gadget screen"
PREFERRED_PROVIDER_virtual/kernel = "linux"
PREFERRED_VERSION_linux ?= "2.6.30"
SERIAL_CONSOLE = "115200 ttyS0 vt100"
USE_VT = "0"
```

Lastly, add the following environmental variables. You can add these to your `~/.profile` if you like.

```
> export BBPATH=/stuff/build:/stuff/openembedded
> export PATH=/stuff/bitbake/bin:$PATH
```

Additionally, you may need to fix your shell. Do the following if OpenEmbedded complains about your shell. To work around this issue call `"sudo dpkg-reconfigure dash"` and select No when it asks you to install dash as `/bin/sh`.

Now we're ready to cross-compile. This is done using BitBake recipes. Fortunately, most commonly used software already has a BitBake recipe. These were provided by the OpenEmbedded git repository we cloned. You can roll your own recipes but support for this is outside the scope of this guide. There is a lot of documentation for this on the BitBake site. Building is done in `/stuff/build` and by default the results are outputted to `/stuff/build/tmp`. This can be modified in your OpenEmbedded configuration file.

```
> cd /stuff/build
> bitbake mysql
```

Assuming BitBake finishes without any errors, you can find your ready to install files at `/stuff/build/tmp/work/armv4t-angstrom-linux-gnueabi/mysql-x.x.x/image`. This directory corresponds to the root of your embedded device and everything needs to be placed accordingly. In our case, we simply symbolically linked the majority of the files. Most of the locale files and many of the binaries don't need to be included if you are looking to save space.

As of this writing, an additional step was performed to get a fully functional MySQL setup. `libncurses` is missing. First we compile `Ncurses` which contains the `libncurses` library.

```
> CC=arm-linux-gcc CFLAGS="-O0 -Wall -W -march=armv4t -msoft-float -
mtune=arm920t -mcpu=arm920t" ./configure --host=arm-linux --with-shared --
```

```
prefix=/usr  
> make
```

The `libncurses.so*` files need to be copied into your MySQL's `/usr/lib/` folder. If a `libread*` is also missing when you run `mysqld`, this library is actually just a symbolic link to corresponding `libncurses.so*` files.

The rest should be standard MySQL setup and is outside the scope of this guide.

References

Parts of this tutorial are taken from the OpenEmbedded wiki - <http://www.openembedded.org>
Ncurses configuration based off commands found on K's Cluttered Loft - <http://www.ailis.de/~k/archives/19-ARM-cross-compiling-howto.html#ncurses>

Copyright

Copyright Anthony Tsang, Ivan Sangala and Ning Wen. Permission given to Brian Fraser to post online.