ARM microcontroller emulation using QEMU

By karibe on February 1, 2014

While I have successfully installed and tested open-source toolchain for ARM based microcontrollers, specifically Freescale chips, teaching embedded systems sometimes requires simulation and emulation. So i think QEMU can fill in this part. In a little while (most probably in the next week) I will be looking at how to set-up QEMU in Ubuntu, build a hello world project for Freescale microcontrollers and execute it in QEMU (without the hardware), and finally debug it using gdb. I will be focussing on the assembly startup codes and probably the UART device.

Most of the groundwork has been covered by balau in a bare-metal hello world arm project so I will use that as reference.

I already covered installation procedures for gnu-arm toolchain in this post.

For starters, Install QEMU in ubuntu

```
sudo apt-get install qemu-system-arm
```

I have had several misc problems with this prebuilt ubuntu binary, so I thought a shortcut would be to build from source.

QEMU build from source

Some things you need as prerequisites: get dependencies and headers for building QEMU in ubuntu. Your source packages must be anabled in sources.list to do this, we usually disable them to speed up apt.

```
sudo apt-get --no-install-recommends -y build-dep qemu
```

Zlib is needed so to avoid that error that “zlib check failed”:

```
sudo apt-get install zlib1g zlib1g-dev
```

I always work from one source location in my home folder:

```
cd ~/src/git
```

May the source be with you:

```
git clone git://git.qemu-project.org/qemu.git
```

If you are operating behind proxy, you need to set up git configuration proxy settings with:

```
git config --global http.proxy http://username:password@proxyserver:port
git config --global https.proxy http://username:password@proxyserver:port
```

Then use the https version of the git URL https://github.com/qemu/qemu

```
git clone https://github.com/qemu/qemu.git
```

You will not be able to get the submodules though, because their URLs are also git protocol and I tried doing the same trick, but I can figure out what their valid URLs for https protocol are. especially the dtc submodule.

Get in there:

```
cd qemu
```

You will get into complications without the dtc submodule so, get it:

```
git submodule update --init dtc
```

Time to configure for the build, I want QEMU for arm, I dont need kvm, and there was an error about qemu-timer, 'timer_settime@@GLIBC_2.2' from librt for glibc 2.17 and above, so there are a few arguments to pass to configure:

```
./configure --extra-ldflags="-ldl -lrt" --disable-kvm --disable-sdl --target-list="arm-softmmu"
```

Build it with four threads:

```
make -j 4
```

Install it

```
sudo make install
```

I have this output at the end:

```
install -m 755 /usr/local/bin/qemu-system-arm
```

An ARM bare metal project for cortex-M0

Bare metal projects are mostly pure C or assembly coded to run directly on the hardware without an operating system. In the embedded world, operating systems are mostly small(relative term) and real-time, like FreeRTOS rather than a full blown OS like Debian. We are not going to be using either of those here. I dont like assembly code much, although its nice for target specific code like startup files so i will keep it there. The rest of the code will be in C.

Lets create a project folder and some source files:

```
mkdir qemutest
cd qemutest/
touch startup.s
touch test.led
touch test.c
```

Then we put some source into startup.s

Launch qemu-system-arm with a binary file to debug

Posted in Embedded systems, Linux | Tagged ARM, embedded systems, emulator, QEMU | Leave a response

Leave a Reply

Name *

Email *

Website

Math Captcha *
forty two − = 32

Comment

Post Comment

« Previous Next