GNU Toolchain

Young W. Lim

October 9, 2014

GNU Toolchains

- Mentor Graphics : CodeSourcery (arm-none-eabi-gcc)
- YAGARTO (arm-none-eabi-gcc)
- Linaro (arm-linux-gnueabi-gcc)

CodeSourcery Cross Compiler

- download from Mentor Graphics arm-2014.05-28-arm-none-eabi-i686-pc-linux-gnu.tar.bz2
- mkdir ~/toolchains
- cd ~/toolchains
- tar -jxf ~/downloads/arm-2014.05-28-arm-none-eabi-i686-pc-linux-gnu.tar.bz2
- export PATH=~/toolchains/arm-2014.05/bin:$PATH

YAGARTO

- download arm-none-eabi.tar.bz2 from http://www.cl.cam.ac.uk/freshers/raspberrypi/tutorials/os/downloads/
- tar xjvf arm-none-eabi.tar.bz2
- export PATH=$PATH:$HOME/arm-2008q3/bin

Linaro ARM Cross Compiler

- sudo apt-get install gcc-arm-linux-gnueabi
- sudo apt-get install libc6-dev-armel-cross

- directory: /usr/arm-linux-gnueabi/
- prefix: arm-linux-gnueabi-

Qemu Emulation Mode

- Full System Emulation
  - prefix qemu-system-
- User Mode Emulation
  - prefix qemu-

Qemu Installation

- Ubuntu
  - sudo apt-get install qemu
- Debian
  - sudo apt-get install qemu
- Fedroa
  - su -c "yum install qemu"

Qemu User Mode Emulation

- sudo apt-get install qemu
- arm-linux-gnueabi-gcc test.c -o test
- vi test.c
- file test
- ./test : not working
- qemu-arm -L /usr/arm-linux-anueabi/ test

QEMU Testing

- bzip2 -d linux-0.2.img.bz2
- gunzip arm-test-0.2.tar.gz
- qemu-system-arm -kernel zImage.integrator -initrd arm_root.img

CodeSourcery Assembly Example

- arm-none-eabi-as -o add.o add.s
- arm-none-eabi-ld -Ttext=0x0 -o add.elf add.o
- arm-none-eabi-nm add.elf
- arm-none-eabi-ld -Ttext=0x20000000 -o add.elf add.o
- arm-none-eabi-nm add.elf
- arm-none-eabi-objcopy -O binary add.elf add.bin
- dd if=/dev/zero of=flash.bin bs=4096 count=4096
- dd if=add.bin of=flash.bin bs=4096 conv=notrunc
- qemu-system-arm -M connex -pflash flash.bin -nographic -serial /dev/null

ABI

- ABI (Application Binary Interface)
  - The abi describes how the compiler should generate the assembler
  - a mapping between low-level concepts in high-level languages
  - the abilities of a specific hw / os platform's machine code
    - ★ The object file format (ELF for EABI)
    - ★ How arguments are passed to functions
    - ★ How many bits are in a char, int, long, etc.
    - ★ How C++ templates get instantiated

EABI (Embedded ABI)

- ELF is an object file format
- EABI is an ABI
- Two versions of the ARM ABI
  - arm-elf
    - ★ generates obj code for some OS which support elf (examle linux ABI). OS will control executing of your program.
  - arm-none-eabi : a newer revision, but also called arm-elf-eabi
    - ★ toolchain generates obj code for microcontrollers or microprocessors
    - ★ this code in the flash or core of microcontroller starts executing after poweron.
    - ★ No OS, extended command set, no usage with shared modules

Reference

[1] http://www.bravegnu.org/gnu-eprog/
[2] http://vineelkumarreddy.wordpress.com/