

Student ID: 406410035

Name: 秦紫頤

Email: chinjoyce30@gmail.com

Lab Title: Remote Debugging

Lab Purpose:

Learn how to do remote debugging on your host machine when the target program is executing on Raspberry Pi. Learn how to compile GDB server.

Lab Procedure:

1. Recompile **cross-toolchain** in the first experiment because we need C++ compiler in our GDB server -> add **--enable-languages=c,c++** when configuring **gcc-4.9.3**
2. Compile GDB server
 - a. `cd /home/joycenerd/myWork/build_gdb/gdb/gdbserver`
 - b. `export CC=arm-linux-gnueabi-hf-gcc`
 - c. `export CXX=arm-linux-gnueabi-hf-g++`
 - d. `export CFLAGS=-static`
 - e. `export CXXFLAGS=-static`
 - f. `./configure --host=arm-linux-gnueabi-hf --target=arm-linux-gnueabi-hf`
 - g. `make`
 - h. There will be **"gdbserver"** executable file in the same directory
3. Generate test files
 - a. Write a simple **test.c** file
 - b. Generate **test_static.exe**: `arm-linux-gnueabi-hf-gcc -static test.c -o test_static.exe`
 - c. Generate **test.exe**: `arm-linux-gnueabi-hf-gcc test.c -o test.exe`
 - d. Generate **test_static_g.exe**: `arm-linux-gnueabi-hf-gcc -static -g test.c -o test_static_g.exe`
 - e. Generate **test_g.exe**: `arm-linux-gnueabi-hf-gcc -g test.c -o test_g.exe`
4. Put files into SD card
 - a. `mkdir exp6`
 - b. Put **"gdbserver"** executable file, `test.c`, `test_static.exe`, `test.exe`, `test_static_g.exe`, `test_g.exe` into `exp6/`
 - c. Plug SD card into the computer
 - d. `sudo mount /dev/sdb2 ~/mmc2`
 - e. `sudo cp -R exp6 ~/mmc2`
5. Power on Raspberry Pi 3 and display it on the external monitor
6. connect target machine (Raspberry Pi 3) and host machine (Ubuntu 16.04) with cross over cable
7. Setup IP
 - a. virtual/host (Ubuntu 16.04): **`ifconfig enx000ec6a47f1d 192.168.1.200 broadcast 192.168.1.255 netmask 255.255.255.0`**
 - b. Target (Raspberry Pi 3): **`ifconfig eth0 192.168.1.100 broadcast 192.168.1.255 netmask 255.255.255.0`**
8. Change directory to `exp6` on both Raspberry Pi 3 and Ubuntu 16.04

9. On Raspberry Pi 3: `./gdbserver 192.168.1.200:1234 ./test_static_g.exe`

10. On Ubuntu 16.04:

- a. `arm-linux-gnueabi-gdb`
- b. file `test_static_g.exe`
- c. target remote `192.168.1.100:1234`
- d. Debug ...

11. In the end, the result will show on Raspberry Pi 3

Problems and Discussion

- Questions

What are the results of `test.exe`, `test_g.exe`, `test_static.exe`, and `test_static_g.exe` on Pi? What are the differences?

1. `test.exe`: can't execute because didn't add `-static` parameter when compiling
2. `test_g.exe`: can't execute because didn't add `-static` parameter when compiling
3. `test_static.exe`: Can't see the corresponding C source code because didn't add `-g` when compiling
4. `test_static_g.exe`: this is the only one that can debug normally on Pi

In conclusion, if you didn't add **“-static”** you can't even execute the program on Pi and if you didn't add **“-g”** you can't see the corresponding source code when debugging then debugging become meaningless

- Discussions

In the experiment handout, there are two ways to set IP. One of them is to modify **/etc/network/interfaces** file. I encountered two problems here.

1. In Ubuntu 16.04: after modifying the file properly and reboot the machine, I see no change and I don't know the reason
2. In Raspberry Pi 3: I can't even modify the file cause no text editor is working

I think the only way is to type **ifconfig** command manually on both machines in my case.