**EE414 Embedded System**

**20180745 – Tung-Duong Mai**

**Lab 1. Cross Development Environment**

1. Purpose:

Get familiar with the cross development system:

Target board: BeagleBone Black with Linux in microSD.

Host computer: PC with Linux, cross compiler, NFS, and minicom.

1. Experiment sequence:
2. Test Ubuntu on PC

Dual-boot my (Windows) PC with Ubuntu 16.04

Then, test out some command in the terminal and check the IP address of my PC (I use WiFi for PC network connection)

1. Install Linux on Beaglebone microSD

Use Debian image, load to microSD and insert to The BeagleBone

Connect the BeagleBone with the PC by USB cable, and connect with internet by LAN connection. (+ for sufficient power, I also connect with power adapter)

1. Configure BeagleBone Debian

Configure the superuser, myself. Network configuration

Update the packages

1. Install Cross-compiler and Test

Install compiler

Test the compiler for the very simple helloes.c program (it simply prints a message)

Use scp to transfer to Bones and check the compiler

1. NFS in Bone-Ubuntu & PC-Ubuntu

Install NFS server on PC and client on BeagleBone

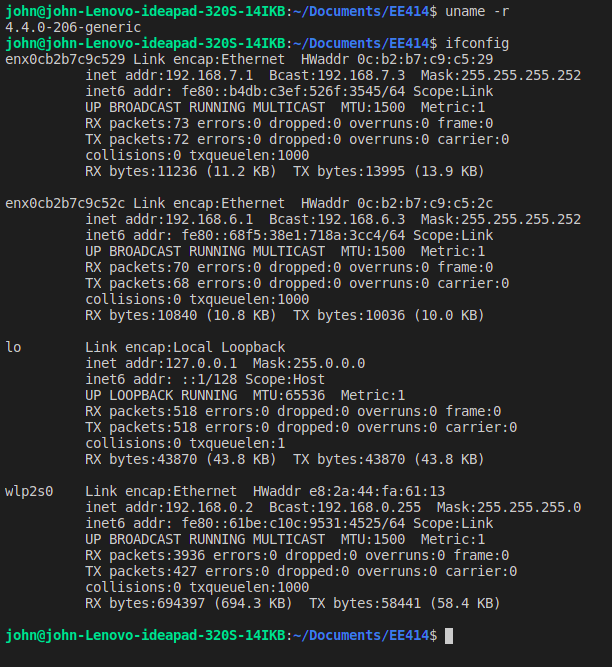
1. Test Makefile

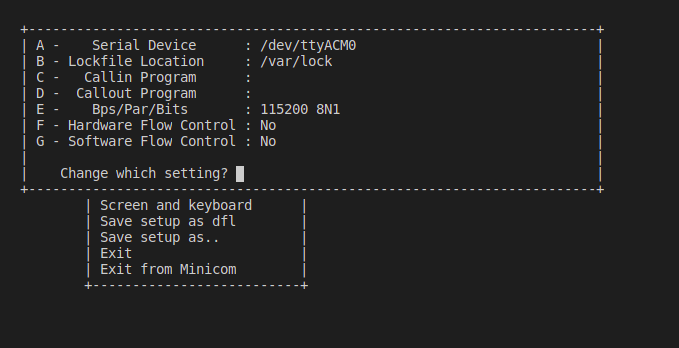
Check native and cross compiler

1. Debug Application on PC

Debug taylor\_ce.c on PC, then cross-compile and test on BeagleBone

1. Experimental results:
2. Test Ubuntu on PC

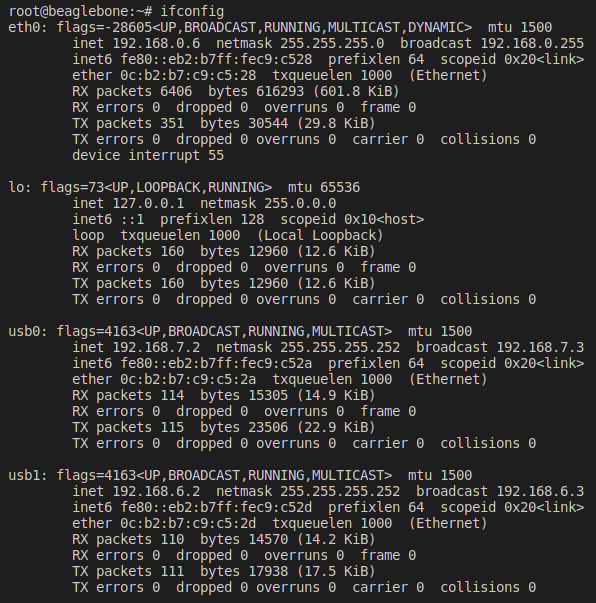


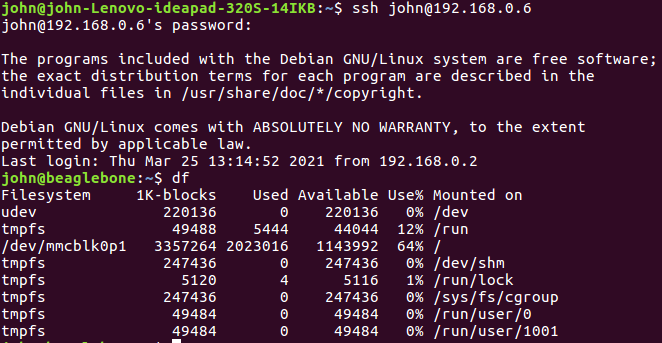


1. Install Linux on Beaglebone

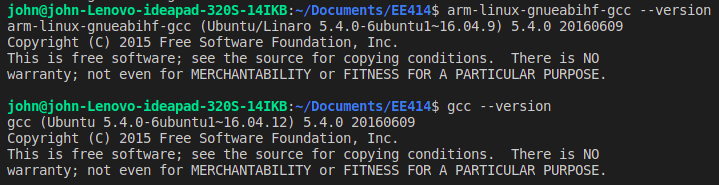
https://lh3.googleusercontent.com/Oy3JarX8-cBxpR3b0yILY54R8AI3ST9heeA9IAV2vMfC789vfy6zo9aRx96Jav_ci8tZwdhQEuSYQI4SJky_qjPGAFwCzxZDOy8pTHMc-mSxJr-UY0GZzWwbPe4USjpHTU2cA7W4

1. Configure BeagleBone Debian





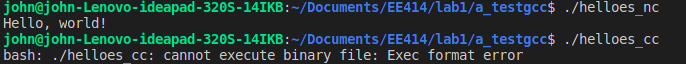
1. Install Cross-compiler and Test



Natively-compiled file (helloes) runs on PC nut not BeagleBone

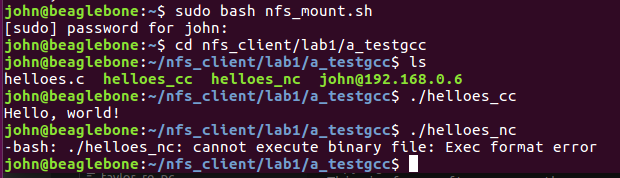
Cross-complied file (helloes.x) runs on BeagleBone, but not PC

PC:



Scp (BeagleBone)

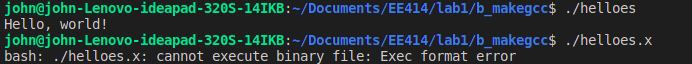
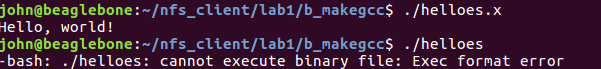
https://lh6.googleusercontent.com/p7pdloP1okdppnxlxDMJwJADp8dxGRjfjEHpUY26DFth00QRMHYIezpz66P_vYBYfz3XgdOqMQP0K6dg0nUgxlP_v5kS29JiIpCv5gX9owE-yW3XNxe9wyEF-WiJ0RsaEtVUqiEl

NFS (BeagleBone)

1. Test Makefile

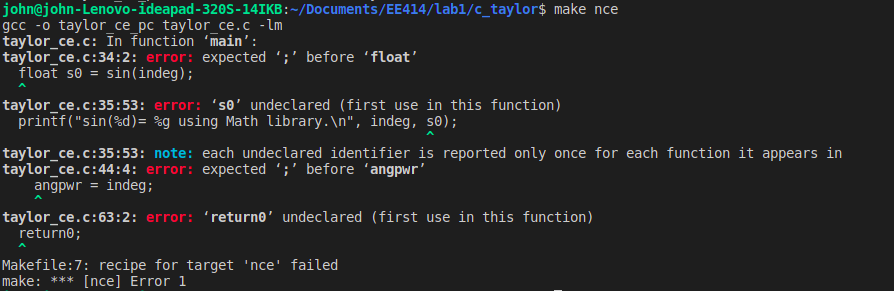
Natively-compiled file (helloes) runs on PC nut not BeagleBone

Cross-complied file (helloes.x) runs on BeagleBone, but not PC

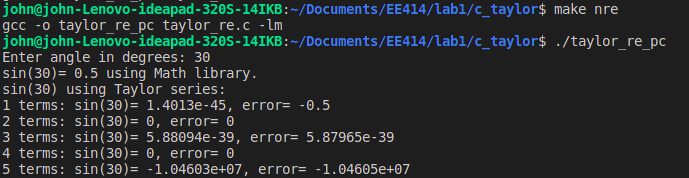


1. Debug Application on PC:

taylor\_ce.c induces compile error

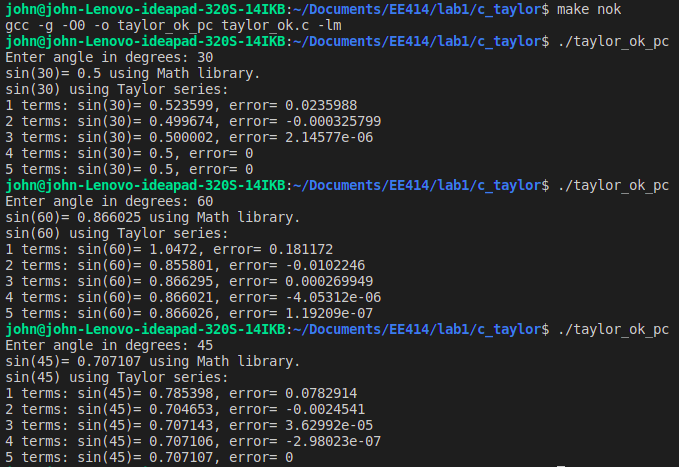


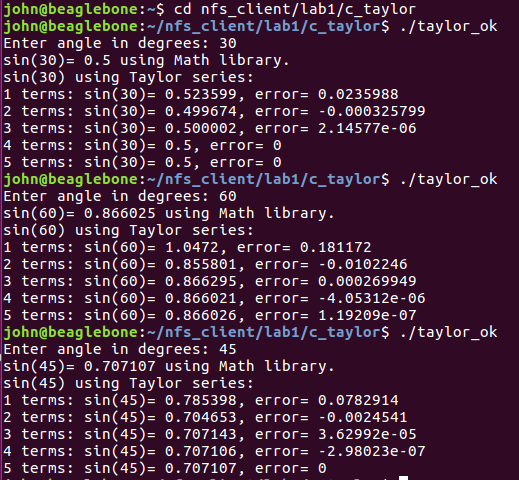
taylor\_re.c (after some debugging) runs without error, but the result is not correct (although result from Math is correct after some debugging)



On PC: taylor\_ok\_pc (natively complied) run well with correct result

On BeagleBone: taylor\_ok (cross compiled) run well with correct result





1. Discussion:

About cross-compilation

- The setting we use in the lab is very handy, as it allows us to share the access to files between PC and BeagleBone. Hence, we can do the code development debugging in PC and cross-compile to run on BeagleBone. This allow us to use the familiar setting with PC (with mouse, keyboard, …) to develop application running on Beaglebones.

- This is called cross-compile because we develop and run the code in two different systems.

- Both minicom and ssh connection works correctly.

About files sharing

- NFS or scp both work correctly. However, I think NFS is more intuitive since it gives me the sentiment that the data are shared and I can work as if the data is locally stored (for scp, I have to explicitly transfer the data).

About running the compiled file

- arm-linux-gnueabihf-gcc generates the Arm-compatible binary file. This couldn’t be run on the PC (x84\_64) architecture because instructions encoding varies among architectures. It could only be executed in the BeagleBone. Similarly, natively-compiled file wont run on BeagleBone.

About the taylor.c code

- Taylor series implementation, although currently works, could be more optimized. For example, we don’t need to calculate s[n] from scratch (we could just add the new term to s[n-1]). Also, we don’t need to restrict the angle to be integer.

For the angle outside of (-180, 180) range, we can bring it to this range (since 360 degree difference produce the same sine) for faster convergence.

1. References:

[1] Getting Started with Beaglebone and Beaglebone Black, http://www.beagleboard.org/Getting%20Started

[2] Beaglebone Rev. A5. System Reference Manual,

http://circuitco.com/support/index.php?title=Beaglebone#Rev\_A5. NOTE. The version of Beaglebone boards in the Lab is Rev. A5. Be sure to download the correct version.

[3] “Embedded Linux Primer”, C. Hallinan, Prentice Hall.

[4] Lab material, EE414 Teaching Staffs, KLMS