學號：405410117
姓名：郭紘安 Email：andy8766kuo@gmail.com

實驗名稱：Cross Toolchain for ARM Linux (Binutils, GCC)

實驗目的：由 source code 開始，自己編譯支援 Raspberry Pi 相關的 cross compiler 、cross assembler 、 cross linker 等工具。

實驗步驟：

1.Build Cross Binutils

```
andy@ubuntu:~/$ mkdir myWORK
andy@ubuntu:~/$ cd myWORK
andy@ubuntu:~/myWORK$ tar -zxvf binutils-2.25.1.tar.gz
andy@ubuntu:~/myWORK$ mkdir build_binutils
andy@ubuntu:~/myWORK$ cd build_binutils/
andy@ubuntu:~/myWORK/build_binutils$ ../binutils-2.25.1/configure
--prefix=/home/andy/WORK/crossgcc1 --target=arm-linux-gnueabihf
andy@ubuntu:~/myWORK/build_binutils$ make
andy@ubuntu:~/myWORK/build_binutils$ make install
```

2.Build a Bare-metal Cross Compiler

```
andy@ubuntu:~/myWORK$ sudo apt-get install libgmp-dev libmpfr-dev libmpc-dev
```

- 安裝 GMP 、 MPFR 、 MPC (Building GCC requires GMP 4.2+, MPFR 2.4.0+ and MPC 0.8.0)

```
andy@ubuntu:~/myWORK$ tar -zxvf gcc-4.9.3.tar.gz
andy@ubuntu:~/myWORK$ mkdir build_gcc1
andy@ubuntu:~/myWORK$ cd build_gcc1
andy@ubuntu:~/myWORK/build_gcc1$ export
PATH="/home/andy/WORK/crossgcc1/bin:$PATH"
```

- Add “/home/andy/WORK/crossgcc1/bin” to PATH

```
andy@ubuntu:~/myWORK/build_gcc1$ ../gcc-4.9.3/configure
--prefix=/home/andy/WORK/crossgcc1 --target=arm-linux-gnueabihf
--enable-languages=c --without-headers --disable-libmudflap
--disable-libatomic --with-arch=armv6 --disable-shared --enable-static
--disable-decimal-float --disable-libgomp --disable-libitm
--disable-libquadmath --disable-libsanitizer --disable-libssp
```

學號：405410117
姓名：郭紘安 Email：andy8766kuo@gmail.com

```
--disable-threads --with-float=hard --with-fpu=vfp
andy@ubuntu:~/myWORK/build_gcc1$ make
andy@ubuntu:~/myWORK/build_gcc1$ make install
```

3.Installing Kernel Headers

```
andy@ubuntu:~/myWORK$ cd linux
andy@ubuntu:~/myWORK/linux$ make headers_install ARCH=arm
INSTALL_HDR_PATH=/home/andy/WORK/sysroot/usr/
```

4.Building GLIBC

```
andy@ubuntu:~/myWORK$ export PATH="/home/andy/WORK/crossgcc1/bin:$PATH"
```

- Add /home/andy/WORK/crossgcc1/bin to the PATH.

```
andy@ubuntu:~/myWORK$ tar -zxvf glibc-2.19.tar.gz
andy@ubuntu:~/myWORK$ mkdir build_eglibc
andy@ubuntu:~/myWORK$ cd build_eglibc
andy@ubuntu:~/myWORK/build_eglibc$ ../glibc-2.19/configure --prefix=/usr
--host=arm-linux-gnueabihf --target=arm-linux-gnueabihf
--with-headers=/home/andy/WORK/sysroot/usr/include --includedir=/usr/include
--enable-add-ons --disable-multilib
andy@ubuntu:~/myWORK/build_eglibc$ make
andy@ubuntu:~/myWORK/build_eglibc$ make install
install_root=/home/andy/WORK/sysroot
```

5.Build Cross Binutils

```
andy@ubuntu:~/myWORK$ mkdir build_binutils2
andy@ubuntu:~/myWORK$ cd build_binutils2
andy@ubuntu:~/myWORK/build_binutils2$ ../binutils-2.25.1/confi
gure --prefix=/home/andy/WORK/crossgcc2
--target=arm-linux-gnueabihf
--with-sysroot=/home/andy/WORK/sysroot
andy@ubuntu:~/myWORK/build_binutils2$ make
andy@ubuntu:~/myWORK/build_binutils2$ make install
```

學號：405410117
姓名：郭紘安 Email：andy8766kuo@gmail.com

6.Build a C ross Compiler

```
andy@ubuntu:~/myWORK$ export
PATH="/home/andy/WORK/crossgcc2/bin:$PATH"
```

- Add /home/andy/WORK/crossgcc2/bin to the PATH.

```
andy@ubuntu:~/myWORK$ mkdir build_gcc2
andy@ubuntu:~/myWORK$ cd build_gcc2
andy@ubuntu:~/myWORK/build_gcc2$ ../gcc-4.9.3/configure
--prefix=/home/andy/WORK/crossgcc2 --target=arm-linux-gnueabihf
--enable-languages=c --with-sysroot=/home/andy/WORK/sysroot
--with-arch=armv6 --with-fpu=vfp --with-float=hard
--disable-libmudflap --enable-libgomp --disable-libssp
--enable-libquadmath --enable-libquadmath-support
--disable-libsanitizer --enable-lto --enable-threads=posix
--enable-target-optspace --with-linker-hash-style=gnu
--disable-nls --disable-multilib --enable-long-long
andy@ubuntu:~/myWORK/build_gcc2$ make
andy@ubuntu:~/myWORK/build_gcc2$ make install
```

7.DEMO: Testing

andy@ubuntu:~/EmbeddedSystem$ arm-linux-gnueabihf-gcc -S test.c

輸入的檔案:test.c

```
#include <stdio.h>
int main(void) {
    printf("Hello! World!\n");
    return 0;
}
```

輸出的檔案:test.s

```
    .arch armv6
    .eabi_attribute 27, 3
    .eabi_attribute 28, 1
    .fpu vfp
    .eabi_attribute 20, 1
    .eabi_attribute 21, 1
    .eabi_attribute 23, 3
    .eabi_attribute 24, 1
    .eabi_attribute 25, 1
    .eabi_attribute 26, 2
    .eabi_attribute 30, 6
    .eabi_attribute 34, 1
    .eabi_attribute 18, 4
    .file    "test.c"
    .section .rodata
    .align   2
.LC0:
```

學號：405410117
姓名：郭紘安 Email：andy8766kuo@gmail.com

```
    .ascii   "Hello! World!\000"
    .text
    .align   2
    .global  main
    .type    main, %function
main:
    @ args = 0, pretend = 0, frame = 0
    @ frame_needed = 1, uses_anonymous_args = 0
    stmfd    sp!, {fp, lr}
    add  fp, sp, #4
    ldr  r0, .L3
    bl   puts
    mov  r3, #0
    mov  r0, r3
    ldmfd    sp!, {fp, pc}
.L4:
    .align   2
.L3:
    .word    .LC0
    .size    main, .-main
    .ident   "GCC: (GNU) 4.9.3"
    .section .note.GNU-stack,"",%progbits
```

執行結果:

andy@ubuntu:~/myWORK$ arm-linux-gnueabihf-gcc test.c -o test -static

透過 COM5 連到 raspberrypi 的執行結果

```
pi@raspberrypi:/mnt/sda1$ ./test
Hello! World!
```

問題與討論：

1. 主要是要產生 ARM 的 runtime library(glibc+header)
2. 為了要產生 ARM 的 runtime library(glibc+header) 需要用 ARM 的 gcc 來編

3.ARM 的 gcc 分兩種 需要 headerfile 跟不需要 headerfile

- 不需要 headerfile 的命名為 crossgcc1
- 需要 headerfile 的命名為 crossgcc2

4.要編譯 ARM 的 gcc 還需要 binutils 所以才會先建立 cross binutils

5.一開始在建立 glibc 時，沒辦法順利執行 configure 產生 Makefile，後來才發現是之前在建立 cross binutils 時

../binutils-2.25.1/configure --prefix=/home/andy/WORK/crossgcc1 --target=arm-linux-gnueabihf

這句的最後少打 f 這個英文字母

學號：405410117
姓名：郭紘安 Email：andy8766kuo@gmail.com

心得：

從這次的實驗中了解了要如何自己編譯支援 Raspberry Pi 的 cross compiler、 cross assembler 、 cross linker 等等，也發現不能做太快，做太快結果漏打了一個 f，之後修改找錯誤反而花更多時間，一步一步慢慢做然侯做對才比較有效率。