

Group 4 - 翁培豪 312551076 / 張紀睿 312551086

以下是Lab5 進行 Cross-compile 的具體步驟：

前置準備

創建安裝目錄：

創建新的目錄分別用來存儲安裝包與編譯後的檔案。

```
mkdir -p ~/cross-compile
```

```
mkdir ~/cross-compile/build
```

下載並解壓縮軟體包：

下載所需的軟體包 (alsa-lib, alsa-utils, zlib, libid3tag, libmad, madplay)，並將它們解壓縮到一個工作目錄中。

```
tar xvf alsa-lib-1.0.26.tar.bz2 -C ~/cross-compile
```

```
tar xvf alsa-utils-1.0.26.tar.bz2 -C ~/cross-compile
```

```
tar xvf zlib-1.2.3.tar.gz -C ~/cross-compile
```

```
tar xvf libid3tag-0.15.1b.tar.gz -C ~/cross-compile
```

```
tar xvf libmad-0.15.1b.tar.gz -C ~/cross-compile
```

```
tar xvf madplay-0.15.2b.tar.gz -C ~/cross-compile
```

設定環境變數：

這將幫助指定交叉編譯工具鏈和安裝位置。您需要設定的環境變數包括 CC(C 編譯器)，CXX(C++ 編譯器)，LD(連結器)，以及 --prefix 參數所指定的安裝目錄。

```
export
```

```
CC=/opt/EmbedSky/gcc-linaro-5.3-2016.02-x86_64_arm-linux-gnueabi  
hf/bin/arm-linux-gnueabi  
hf-gcc
```

```
export
```

```
CXX=/opt/EmbedSky/gcc-linaro-5.3-2016.02-x86_64_arm-linux-gnueabi  
hf/bin/arm-linux-gnueabi  
hf-g++
```

```
export
```

```
LD=/opt/EmbedSky/gcc-linaro-5.3-2016.02-x86_64_arm-linux-gnueabi  
hf/bin/arm-linux-gnueabi  
hf-ld
```

```
export PREFIX=~/cross-compile/build
```

alsa-lib

進入 **alsa-lib** 目錄：

```
cd ~/cross-compile/alsa-lib-1.0.26
```

配置 **alsa-lib**：

```
./configure --host=arm-linux-gnueabi \
--enable-shared --disable-python \
--with-configdir=/usr/local/share/alsa \
--with-plugindir=/usr/local/lib/alsa_lib \
CC=$CC CXX=$CXX LD=$LD
```

編譯並安裝 **alsa-lib**：

```
make
sudo make install
```

alsa-utils

切換到 **alsa-utils** 的目錄：

```
cd ~/cross-compile/alsa-utils-1.0.26
```

配置 **alsa-utils**：

```
./configure --host=arm-linux-gnueabi \
--prefix=$PREFIX \
CFLAGS="-I$PREFIX/include" \
LDFLAGS="-L$PREFIX/lib -lasound" \
--disable-alsamixer --disable-xmlto \
--with-alsa-inc-prefix=$PREFIX/include \
--with-alsa-prefix=$PREFIX/lib \
CC=$CC CXX=$CXX LD=$LD
```

編譯並安裝 **alsa-utils**：

```
make
sudo make install
```

zlib

切換到 **zlib** 的目錄：

```
cd ~/cross-compile/zlib-1.2.3
```

配置 **zlib** 並加入 **-fPIC**：

```
CFLAGS="-O3 -fPIC" ./configure --prefix=$PREFIX
```

修改 **Makefile** 以使用交叉編譯工具：

```
sed -i 's/CC=gcc/CC=arm-linux-gnueabi-gcc/g; \
s/LDSHARED=gcc -shared -Wl,-soname,libz.so.1/LDSHARED=arm-linux-gnueabi-gcc \
-shared -Wl,-soname,libz.so.1/g; \
s/CPP=gcc -E/CPP=arm-linux-gnueabi-gcc -E/g; \
s/AR=ar rc/AR=arm-linux-gnueabi-ar rc/g; \
s/RANLIB=ranlib/RANLIB=arm-linux-gnueabi-ranlib/g' Makefile
```

編譯並安裝 **zlib**：

```
make
sudo make install
```

libid3tag

切換到 **libid3tag** 目錄：

```
cd ~/cross-compile/libid3tag-0.15.1b
```

配置 **libid3tag**：

```
./configure --host=arm-linux-gnueabi \
--prefix=$PREFIX \
CPPFLAGS=-I$PREFIX/include \
LDFLAGS=-L$PREFIX/lib \
CC=$CC
```

編譯並安裝 **libid3tag**：

```
make
sudo make install
```

libmad

進入 **libmad** 的目錄：

```
cd ~/cross-compile/libmad-0.15.1b
```

配置 **libmad**：

```
./configure --host=arm-linux-gnueabi \
    --prefix=$PREFIX \
    CPPFLAGS="-I$PREFIX/include" \
    LDFLAGS="-L$PREFIX/lib" \
    CC=$CC
```

編輯 **Makefile** 以移除 **-fforce-mem** 選項(因為新版本的**GCC**不再支持)：

```
sed -i 's/-fforce-mem//g' Makefile
```

將**fixed.h**中的下方段落替換：

```
# define MAD_F_MLN(hi, lo) \
    asm ("rsbs %0, %2, #0\n\t" \
        "rsc %1, %3, #0" \
        : "=r" (lo), "=r" (hi) \
        : "0" (lo), "1" (hi) \
        : "cc")
```

替換為：

```
#ifdef __thumb__
/* In Thumb-2, the RSB-immediate instruction is only allowed with a zero
operand. If needed this code can also support Thumb-1
(simply append "s" to the end of the second two instructions). */
# define MAD_F_MLN(hi, lo) \
    asm ("rsbs %0, %0, #0\n\t" \
        "sbc %1, %1, %1\n\t" \
        "sub %1, %1, %2" \
        : "+&r" (lo), "&r" (hi) \
        : "r" (hi) \
        : "cc")
#else /* ! __thumb__ */
# define MAD_F_MLN(hi, lo) \
    asm ("rsbs %0, %2, #0\n\t" \
        "rsc %1, %3, #0" \
        : "=r" (lo), "=r" (hi) \
        : "&r" (lo), "=r" (hi) \
```

```
        : "0" (lo), "1" (hi) \
        : "cc")
#endif /* __thumb__ */
```

編譯並安裝 **libmad**:

```
make
sudo make install
```

madplay

進入 **madplay** 目錄:

```
cd ~/cross-compile/madplay-0.15.2b
```

配置 **madplay**:

```
./configure --host=arm-linux-gnueabi --prefix=$PREFIX \
CPPFLAGS=-I$PREFIX/include LDFLAGS=-L$PREFIX/lib \
CC=$CC CXX=$CXX LD=$LD
```

編譯與安裝 **madplay**:

```
make
sudo make install
```

在虛擬機上複製文件到 SD 卡

在 **SD** 卡上創建三個資料夾:

bin、lib 和 alsa。

複製執行檔和庫(**tony**是本組實驗用的虛擬機名稱):

從 /home/tony/cross-compile/build/bin 將 aplay 和 madplay 複製到 SD 卡的 bin 資料夾。
從 /home/tony/cross-compile/build/lib 將所有 .so 文件 (不包括符號鏈接) 複製到 SD 卡的 lib 資料夾。

複製 **ALSA** 配置:

從 /usr/local/share/alsa 將所有文件和資料夾複製到 SD 卡的 alsa 資料夾。

在開發板上的操作

從 **SD** 卡移動文件：

將 SD 卡插入您的 ARM 目標板。

將 SD 卡上 bin 資料夾中的 aplay 和 madplay 移動到目標板的 /usr/bin。

將 SD 卡上 lib 資料夾中的 .so 文件移動到目標板的 /usr/lib。

將 SD 卡上 alsa 資料夾中的所有文件和資料夾移動到目標板的 /usr/local/share/alsa。

建立對應的符號鏈接：

在目標板上的 /usr/lib 目錄中為 libid3tag.so.0.3.0、libmad.so.0.2.1 和 libasound.so.2.0.0。

```
ln -sf /usr/lib/libid3tag.so.0.3.0 /usr/lib/libid3tag.so.0
```

```
ln -sf /usr/lib/libmad.so.0.2.1 /usr/lib/libmad.so.0
```

```
ln -sf /usr/lib/libasound.so.2.0.0 /usr/lib/libasound.so.2
```

```
ldconfig
```

播放音訊：

```
madplay -o wav:- music.mp3 | aplay
```

參考

- <https://chat.openai.com/>
- <https://note.drx.tw/2008/04/command.html>
- <https://www.twblogs.net/a/5e6b933dbd9eee211685f17d>
- https://blog.csdn.net/weixin_44175439/article/details/120186387
- <https://www.cnblogs.com/chd-zhangbo/p/5270290.html>
- <https://groups.google.com/g/linux.debian.bugs.dist/c/d1lBFkrKfRM>