

Android原生(Native)C开发：编译选项小结 (2009-05-04 09:53:33)



标签： android toolchain makefile 动态库 编译 选项 参数 分类： Android

在Android的所有源码中,只有最顶层的目录有一个Makefile,其它所有项目都是调用这个文件进行编译。如果用“showcommands”打出详细编译信息,你会看到大量的编译选项,其实根本不需要这么多,做如下步骤:

- 将所有头文件复制到 toolchain 的 include 目录,可减少大量的 -I (头文件查找目录)选项,具体请参见: [Android原生\(Native\)C开发之八: Toolchain环境搭建篇](#);
- 更多选项查看 `$android_src/build/core/combo/linux-arm.mk`, 这是一个为Android准备的编译选项文件,基本上大部分的项目的编译都依赖此文件。

做完了上面两步,以下是我整理的一些编译选项:

- 编译普通C文件, CFLAGS如下:

```
CFLAGS = -O2 -include AndroidConfig.h
CFLAGS+= -msoft-float -fpic \
-ffunction-sections \
-funwind-tables \
-fstack-protector \
-fno-short-enums
```

- 编译普通CPP文件, CXXFLAGS如下 (与CFLAGS比,只是多了粗体的几个选项):

```
CFLAGS = -O2 -include AndroidConfig.h
CFLAGS+= -msoft-float -fpic \
-ffunction-sections \
-funwind-tables \
-fstack-protector \
-fno-short-enums \
-fno-rtti \
-fstrict-aliasing \
-fno-exceptions \
-fomit-frame-pointer
```

- 可执行程序链接选项 (除了如下选项,还要再加上项目所需的库,如-lm, -lstdc++等):

```
LIB_DIR = /cygdrive/e/toolchain/arm-eabi/lib
CRT_OBJS= $(LIB_DIR)/crtbegin_dynamic.o $(LIB_DIR)/crtend.o
LDFLAGS = -s -nostdlib -Bdynamic \
LDFLAGS+= -Wl,-T,armelf.x \
-Wl,-dynamic-linker,/system/bin/linker
-Wl,--gc-sections \
-Wl,-z,nocopyreloc \
-lc $(CRT_OBJS)
```

- 动态库链接选项 (编译时将 libtest.so 替换成你的库名,再加上项目所需的库):

```
SDFLAGS = -s -nostdlib \
-Wl,-soname,libtest.so \
-Wl,-shared,-Bsymbolic \
-Wl,-T,armelf.xsc \
-Wl,--gc-sections \
-lc -lm
```

基本上,加上这些编译选项,你就可以写出自己的Makefile来编译项目,我将这些编译选项加上,已成功编译SDL、jpeg、libpng等开源项目, [更多的编译选项请参见\\$android_src/build/core/combo/linux-arm.mk](#)。