



链滴

# Unity 编译 mono 库

作者: [bingqilin89757](#)

原文链接: <https://ld246.com/article/1537455295351>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



## 环境说明

- mac osx 10.12.6
- unity 2017.4.6f1

## 步骤

- 安装相关的工具

```
brew install autoconf automake libtool pkg-config
```

- 克隆目录并跳转到目录下切换unity版本分支

```
git clone https://github.com/Unity-Technologies/mono.git
```

```
cd mono
```

```
git checkout unity-2017.4
```

- 初始化子模块
- 赋权并执行编译

```
./external/buildscripts/build_runtime_android.sh
```

```
1. ./autogen.sh (m4)
rmv5 --forcedefaultbuilddeps=1
>>> Mono checkout = /Users/bob/workspace/other/mono
>> System Info :
Darwin igg-bobdeMac-mini.local 16.7.0 Darwin Kernel Version 16.7.0: Thu Jun 15 1
7:36:27 PDT 2017; root:xnu-3789.70.16~2/RELEASE_X86_64 x86_64
>>> Mono Revision = 16e0200d3f1a3ae1d6ec548d14e527af267daafd

>>> Build Scripts Revision = 16e0200d3f1a3ae1d6ec548d14e527af267daafd

>>> Existing Mono =
>>> Mono Arch = i386
>>> No existing mono supplied. Checking for external...
>>> No external build deps found. Might as well try to check them out. If it f
ails, we'll continue and trust mono is in your PATH
>>> Checking out mono build dependencies to : /Users/bob/workspace/other/mono/..
/./mono-build-deps/build
>>> Cloning https://ono.unity3d.com/unity-extra/mono-build-deps at /Users/bob/wo
rkspace/other/mono/../../mono-build-deps/build
>>> No external mono found. Trusting a new enough mono is in your PATH.

>>> Building autoconf, texinfo, automake, and libtool if needed...
failed to chdir to external directory
Failed building mono for armv5
→ mono git:(unity-2017.4-mbe) ✘ ./external/buildscripts/build_runtime_android.s
h
```

- 遇到错误,赋权并运行 autogen.sh

[./autogen.sh](#)

```
1. bob@igg-bobdeMac-mini: ~/workspace/other/mono (zsh)
checking curses.h presence... yes
checking for curses.h... yes
checking for term.h... yes
checking for termios.h usability... yes
checking for termios.h presence... yes
checking for termios.h... yes
checking for socklen_t... yes
checking for array element initializer support... yes
checking for trunc... yes
checking for aintl in -lsurmath... no
checking for execvp... yes
checking if usage of random device is requested... yes
checking for random device... yes
checking if big-arrays are to be enabled... no
checking for dtrace... /usr/sbin/dtrace
checking sys/sdt.h usability... yes
checking sys/sdt.h presence... yes
checking for sys/sdt.h... yes
checking for ranlib that supports -no_warning_for_no_symbols option... yes
checking malloc.h usability... no
checking malloc.h presence... no
checking for malloc.h... no
checking for cmake... no
configure: error: "cmake not found"
→ mono git:(unity-2017.4-mbe) x
```

- 遇到错误,安装cmake

brew install cmake

- 再次运行autogen.sh,经过漫长的刷屏编译



```
1. bob@igg-bobdeMac-mini: ~/workspace/other/mono (zsh)
BigArrays:      no
DTrace:        yes
LLVM Back End: no (dynamically loaded: no)
Interpreter:   no

Libraries:
.NET 4.x:       yes
Xamarin.Android: no
Xamarin.iOS:   no
Xamarin.WatchOS: no
Xamarin.TVOS:  no
Xamarin.Mac:   no
Windows AOT:   no
Orbis:         no
Unity JIT:     default
Unity AOT:     default
Test profiles: AOT Full (no), AOT Hybrid (no)
JNI support:   IKVM Native
libgdiplus:   assumed to be installed
zlib:         system zlib
BTLS:         yes (x86_64)

Now type `make` to compile
→ mono git:(unity-2017.4-mbe) x
```

- 输入make进行编译

make

```
1. bob@igg-bobdeMac-mini: ~/workspace/other/mono (zsh)
Merging: mono-api-type.html
Merging: mono-api-types.html
Merging: mono-api-unsorted.html
Merging: mono-api-utils.html
Merging: mono-api-vm.html
Merging: mono-api-wapi.html
touch deploy/.stamp
/Applications/Xcode.app/Contents/Developer/usr/bin/make -f ./docs.make topdir=./
../mcs srcdir=. mono-file-formats.tree
MDOC [net_4_x] mono-file-formats.tree
/Applications/Xcode.app/Contents/Developer/usr/bin/make -f ./docs.make topdir=./
../mcs srcdir=. mono-tools.tree
MDOC [net_4_x] mono-tools.tree
mkdir -p deploy
cp -f ./api-style.css deploy
/Applications/Xcode.app/Contents/Developer/usr/bin/make -f ./docs.make topdir=./
../mcs srcdir=. monoapi.tree
MDOC [net_4_x] monoapi.tree
Warning: File `DoesNotExist' referenced in TOC but it doesn't exist. It will be
ignored.
Making all in acceptance-tests
make[2]: Nothing to be done for `all'.
Making all in llvm
make[2]: Nothing to be done for `all'.
→ mono git:(unity-2017.4-mbe) x
```

- 再次执行build\_runtime\_android.sh编译报错

```
checking for arm-eabi-linux-gcc... /Users/bob/workspace/other/android-ndk-r10e/toolchains/arm-linux-androideabi-4.8/prebuilt/darwin-x86_64/bin/arm-linux-androideabi-gcc --sysroot=/Users/bob/workspace/other/android-ndk-r10e/platforms/android-ndk-r10e/arm-linux-androideabi
checking for arm-eabi-linux-gcc... (cached) /Users/bob/workspace/other/android-ndk-r10e/toolchains/arm-linux-androideabi-4.8/prebuilt/darwin-x86_64/bin/arm-linux-androideabi-gcc --sysroot=/Users/bob/workspace/other/android-ndk-r10e/platforms/android-ndk-r10e/arm-linux-androideabi
checking whether the C compiler works... no
configure: error: in '/Users/bob/workspace/other/mono':
configure: error: C compiler cannot create executables
See `config.log' for more details
Configure FAILED!
```

- 查看日志

```
/Users/bob/workspace/other/android-ndk-r10e/toolchains/arm-linux-androideabi-4.8/prebuilt/darwin-x86_64/bin/arm-linux-androideabi-ld: error: cannot find -lkrait-signal-handler
collect2: error: ld returned 1 exit status
configure:4557: $? = 1
```

- 怀疑是ndk版本问题导致开始分析代码

```

99 ..... echo "Erasing builds folder to make sure we start with a clean slate"
100 ..... rm -rf builds
101 fi
102
103 function clean_build_krait_patch
104 {
105 ..... local KRAIT_PATCH_REPO="git://github.com/Unity-Technologies/krait-signal-handler.git"
106 ..... if [ ${UNITY_THISISABUILDACHINE:+1} ]; then
107 .....     echo "Trusting TC to have cloned krait patch repository for us"
108 ..... elif [ -d "$KRAIT_PATCH_PATH" ]; then
109 .....     echo "Krait patch repository already cloned"
110 ..... else
111 .....     git clone --branch "master" "$KRAIT_PATCH_REPO" "$KRAIT_PATCH_PATH"
112 ..... fi
113 ..... (cd "$KRAIT_PATCH_PATH" && ./build.pl)
114 }

```

定位到实际上是build.pl在搞鬼,mono这里编译需要的ndk版本是10e,但是krait-signal-handler要求ndk版本是13b,注释掉krait-signal-handler工程中的PrepareAndroidSDK.pm中PrepareNDK代码

```

235 ..... }
236 ..... if ($sdk)
237 ..... {
238 .....     PrepareSDK($sdk);
239 ..... }
240 ..... print "\n";
241 ..... }
242
243 if ($ndk)
244 {
245 ..... print "Installing NDK '$ndk':\n";
246 ..... if (!$ENV{$NDK_ROOT_ENV})
247 ..... {
248 .....     $ENV{$NDK_ROOT_ENV} = catfile($HOME, "android-ndk-auto-" . $ndk);
249 .....     print "\t\t$NDK_ROOT_ENV not set; using $ENV{$NDK_ROOT_ENV} instead\n";
250 ..... }
251
252 ..... # PrepareNDK($ndk);
253 ..... print "\n";
254 ..... }
255 ..... }

```

- 再次执行build\_runtime\_android.sh 编译成功
- 解决编译的so包大小问题

经过对比发现,编译出来的so很大,查了资料发现是debug版本,需要修改两个编译的sh

build\_runtime\_android.sh

修改

```

CFLAGS="\
-DANDROID -DPLATFORM_ANDROID -DLINUX -D_linux_\
-DHAVE_USR_INCLUDE_MALLOC_H -DPAGE_SIZE=0x1000_\
-D_POSIX_PATH_MAX=256 -DS_IWRITE=S_IWUSR_\
-DHAVE_PTHREAD_MUTEX_TIMEDLOCK_\
-fpic -O2 -funwind-tables_\
-ffunction-sections -fdata-sections"

```

为了加快打包速度,注释掉

```
# clean_build "$CCFLAGS_ARMv5_CPU" "$LD_FLAGS_ARMv5" "$OUTDIR/armv5"  
# clean_build "$CCFLAGS_ARMv6_VFP" "$LD_FLAGS_ARMv5" "$OUTDIR/armv6_vfp"
```

build\_runtime\_android\_x86.sh

修改

```
CFLAGS="\  
-DANDROID -DPLATFORM_ANDROID -DLINUX -D_linux_ \  
-DHAVE_USR_INCLUDE_MALLOC_H -DPAGE_SIZE=0x1000 \  
-D_POSIX_PATH_MAX=256 -DS_IWRITE=S_IWUSR \  
-DHAVE_PTHREAD_MUTEX_TIMEDLOCK \  
-fpic -O2 \  
-ffunction-sections -fdata-sections"
```

- 重新编译即可,编译生成库文件在builds/embedruntimes/android目录下

