目录

iOS逆向开发：MonkeyDev调试

- 最新版本：`v1.0.1`
- 更新时间：`20241007`

简介

整理iOS逆向开发中动态调试和插件tweak开发都会涉及到的工具MonkeyDev。先是概览；然后介绍环境搭建，包括初始化安装MonkeyDev，以如何及用Xcode+MonkeyDev去动态调试YouTube的ipa的过程；然后介绍MonkeyDev内部包含的内容，class-dump、LLDBTools等；然后总结心得，包括内部脚本逻辑、项目代码结构。

源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

HonKit源码

- crifan/ios_re_monkeydev_debug: iOS逆向开发：MonkeyDev调试

如何使用此HonKit源码去生成发布为电子书

详见：crifan/honkit_template: demo how to use crifan honkit template and demo

在线浏览

- iOS逆向开发：MonkeyDev调试 book.crifan.org
- iOS逆向开发：MonkeyDev调试 crifan.github.io

离线下载阅读

- iOS逆向开发：MonkeyDev调试 PDF
- iOS逆向开发：MonkeyDev调试 ePub
- iOS逆向开发：MonkeyDev调试 Mobi

版权和用途说明

鸣谢

感谢我的老婆**陈雪**的包容理解和悉心照料，才使得我 `crifan` 有更多精力去专注技术专研和整理归纳出这些电子书和技术教程，特此鸣谢。

其他

作者的其他电子书

本人 `crifan` 还写了其他 `150+` 本电子书教程，感兴趣可移步至：

crifan/crifan_ebook_readme: Crifan的电子书的使用说明

关于作者

关于作者更多介绍，详见：

关于CrifanLi李茂 – 在路上

MonkeyDev概览

iOS逆向开发期间，其中常会涉及到动态调试和写tweak插件，其中有个很好用的工具就是：`MonkeyDev`

- `MonkeyDev`
  - 是什么：iOS逆向开发的成套工具
  - 概述：**iOSOpenDev的升级版** = 集成XCode和其他各种工具的更强的集成环境
  - 一句话描述：一个基于Xcode模块技术快速开发越狱和非越狱插件的工具，可以自动完成逆向中的固定步骤，一键集成非越狱插件，大大提升逆向分析和开发效率
  - 形式：Xcode的一个插件，可以新建MonkeyDev的相关不同类型的项目，做相关的逆向开发
  - 典型的用途
    - 砸壳出ipa后，用MonkeyDev+Xcode去动态调试
    - 用MonkeyDev去写（iPhone越狱后的）tweak插件
  - 主要包含模块
    - `Logos Tweak`
      - 使用theos提供的logify.pl工具将.*xm*文件转成.mm文件进行编译，集成了CydiaSubstrate，可以使用MSHookMessageEx和MSHookFunction来Hook OC函数、C/C++函数或指定地址
    - `CaptainHook Tweak`
      - 使用CaptainHook提供的头文件进行OC函数的Hook，以及属性的获取
    - `Command-line Tool`
      - 可以直接创建运行于越狱设备的命令行工具
    - `MonkeyApp`
      - 自动给第三方应用集成Reveal、Cycript和注入dylib的模块，支持调试dylib和第三方应用，支持Pod给第三方应用集成SDK，只需要准备一个砸壳后的ipa或者app文件即可
    - `MonkeyPod`
      - 将自动开发的非越狱插件制造成Pod以供其它人通过pod的方法来使用
    - `MonkeyAppMac`
      - 针对Mac逆向开发的模块，可以自动集成substitute，注入以及符号还原工作

MonkeyDev vs iOSOpenDev

- MonkeyDev vs iOSOpenDev
  - MonkeyDev比iOSOpenDev）多出一些更加有用的参数：
    - MonkeyDevDevicePassword
      - 默认值：`alpine`
    - MonkeyDevTheosPath
      - 默认值：`/opt/theos`
    - MonkeyDevKillProcessOnInstall
      - 默认值：`SpringBoard`

官方资料

- 官方资料

- Github
  - AloneMonkey/MonkeyDev: CaptainHook Tweak、Logos Tweak and Command-line Tool、Patch iOS Apps, Without Jailbreak.
    - https://github.com/AloneMonkey/MonkeyDev
  - wiki
    - https://github.com/AloneMonkey/MonkeyDev/wiki
      - 开始使用
      - 非越狱App集成
  - 代码
    - MonkeyDev/bin/md at master · AloneMonkey/MonkeyDev
      - `export PATH=/opt/MonkeyDev/bin:$MonkeyDevTheosPath/bin:/usr/local/bin:/usr/bin:/usr/sbin:/bin:/sbin:$PATH`
  - 相关
    - AloneMonkey/MonkeyDev-Xcode-Templates: MonkeyDev-Xcode-Templates
      - https://github.com/AloneMonkey/MonkeyDev-Xcode-Templates
- Blog
  - https://blog.alonemonkey.com/
    - iOSOpenDev修改版MonkeyDev

环境搭建

TODO：

- 【记录】研究YouTube广告拦截导致视频从头播放的原因：XCode+MonkeyDev动态调试
- 【已解决】Xcode调试越狱iPhone6中的YouTube
- 【记录】恢复iOS逆向Xcode调试YouTube的开发环境
- 【记录】恢复自己Mac的iOS逆向开发环境
- 【已解决】自己Mac中恢复和重建Xcode的MonkeyDev开发环境
- 【未解决】用XCode和MonkeyDev去调试iOS抖音app

初始化MonkeyDev开发环境

> ⚠ 安装路径/opt不能变
>
> 后续的 `MonkeyDev` 、 `theos` 等的安装路径选择，虽然按道理可以自定义，但是此处内部很多脚本貌似只支持固定的默认的路径
>
> 所以，只能安装到默认的固定路径：
>
> - `/opt/MonkeyDev`
> - `/opt/theos`
>
> 而不能轻易改变路径，否则后续会出现很多诡异的问题

初始化搭建MonkeyDev环境=初始化安装MonkeyDev：

- 下载theos

```
sudo git clone --recursive https://github.com/theos/theos.git /opt/theos
```

- 下载MonkeyDev（到固定位置： `/opt/MonkeyDev` ）

```
sudo git clone https://github.com/AloneMonkey/MonkeyDev.git /opt/MonkeyDev
```

- 本地运行脚本去安装

```
cd MonkeyDev/bin
sudo bash md-install
```

搭建好的环境，对应目录的文件

```
 crifan@licrifandeMacBook-Pro ~ ll /opt/MonkeyDev
total 88
drwxr-xr-x   7 root  wheel   224B  6 28 22:01 Frameworks
-rw-r--r--   1 root  wheel    34K  6 28 22:26 LICENSE
drwxr-xr-x   3 root  wheel    96B  6 28 22:01 Librarys
drwxr-xr-x   4 root  wheel   128B  6 28 22:01 MFrameworks
-rw-r--r--   1 root  wheel   1.7K  6 28 22:26 README.md
drwxr-xr-x   3 root  wheel    96B  6 28 22:01 Resource
drwxr-xr-x   4 root  wheel   128B  6 28 22:01 Tools
drwxr-xr-x  12 root  wheel   384B  6 28 22:07 bin
-rw-r--r--   1 root  wheel   802B  6 28 22:26 change.log
drwxr-xr-x   4 root  wheel   128B  6 28 22:01 include
drwxr-xr-x  14 root  wheel   448B  6 28 22:03 templates

 crifan@licrifandeMacBook-Pro ~ ll /opt/theos
total 112
-rw-r--r--   1 root  wheel   5.1K  6 28 21:59 CODE_OF_CONDUCT.md
-rw-r--r--   1 root  wheel    35K  6 28 21:59 LICENSE.md
-rw-r--r--   1 root  wheel   1.0K  6 28 21:59 Prefix.pch
```

```
-rw-r--r--   1 root  wheel   3.1K  6 28 21:59 README.md
drwxr-xr-x  17 root  wheel   544B  6 28 21:59 bin
drwxr-xr-x   3 root  wheel    96B  6 28 21:59 extras
drwxr-xr-x   3 root  wheel    96B  6 28 21:59 include
drwxr-xr-x   3 root  wheel    96B  6 28 21:59 lib
drwxr-xr-x  28 root  wheel   896B  6 28 21:59 makefiles
drwxr-xr-x   3 root  wheel    96B  6 28 21:59 mod
-rw-r--r--   1 root  wheel   657B  6 28 21:59 package.json
drwxr-xr-x   3 root  wheel    96B  6 28 21:59 sdks
drwxr-xr-x   3 root  wheel    96B  6 28 21:59 templates
drwxr-xr-x   3 root  wheel    96B  6 28 21:59 toolchain
drwxr-xr-x   8 root  wheel   256B  6 28 21:59 vendor
```

常见问题

此处整理MonkeyDev环境初始化期间的常见问题。

curl: (7) Failed to connect to raw.githubusercontent.com port 443: Connection refused

```
curl: (7) Failed to connect to raw.githubusercontent.com port 443: Connection refused
Failed to download https://raw.githubusercontent.com/AloneMonkey/frida-ios-dump/3.x/dump.py to /opt/MonkeyDev/bin/dump.py
```

解决办法：

另外单独下载 `frida-ios-dump` ：

```
git clone https://github.com/AloneMonkey/frida-ios-dump.git
```

然后把其中的 `dump.py` 和 `dump.js` 拷贝到 `/opt/MonkeyDev/bin/`

->

- `/opt/MonkeyDev/bin/dump.py`
- `/opt/MonkeyDev/bin/dump.js`

Failed to extract /xxx/md-install.gvGnDuMp/file.tar.gz to

```
Failed to extract /var/folders/zz/zyxvpxvq6csfxvn_n0000000000000/T/md-install.gvGnDuMp/file.tar.gz to /var/folders/zz/zyxvpxvq6csfxvn_n0000000000000/T/md-install.KQllUKhp
```

解决办法：

自己新建一个临时目录：

```
mkdir -p /tmp/md_install/tempdirs
```

改 `bin/md-install` 为：

```
# export tempDirsFile="`mktemp -d -t $scriptName`/tempdirs"
export tempDirsFile="/tmp/md_install/tempdirs"
```

Failed to echo into

错误现象：

```
line 82行: Failed to echo into
```

解决办法：

注释掉

```
    # echo "$tempDir" >> "$tempDirsFile" || \
    #     panic $? "Failed to echo into $tempDirsFile"
```

File /xxx/Specifications/MacOSX Package Types.xcspec not found

```
➜  bin sudo bash md-install
...
File /Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/Lib
rary/Xcode/Specifications/MacOSX Package Types.xcspec not found
```

解决办法：

- Xcode <13
  - 背景：存在 `MacOSX Package Types.xcspec`，只是路径不对
  - 解决办法：改动路径或换用软链接
- Xcode 13+
  - 背景：不存在 `MacOSX Package Types.xcspec`（和 `MacOSX Product Types.xcspec`），所以要去网上下载后，再去：改动路径或换用软链接
    - 下载 `MacOSX Package Types.xcspec` 和 `MacOSX Product Types.xcspec`
      - qbs/share/qbs/modules/bundle at master · qbs/qbs中，下载
        - https://github.com/qbs/qbs/blob/master/share/qbs/modules/bundle/MacOSX-Package-Types.xcspec
          - 保存为：`MacOSX Package Types.xcspec`
        - https://github.com/qbs/qbs/blob/master/share/qbs/modules/bundle/MacOSX-Product-Types.xcspec
          - 保存为：`MacOSX Product Types.xcspec`
      - 拷贝到（旧版Xcode中对应的）目录：`/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/Library/Xcode/PrivatePlugIns/IDEOSXSupportCore.ideplugin/Contents/Resources`

然后继续去操作：

- 【推荐】方法1：使用软链接

```
sudo ln -s /Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Develop
er/Library/Xcode/PrivatePlugIns/IDEOSXSupportCore.ideplugin/Contents/Resources /Applica
tions/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/Library/Xcode/Sp
ecifications
```

- 方法2：（修改 `md-install` 脚本）改动路径

修改 `/opt/MonkeyDev/bin/md-install`

修改路径，改为：

```
# macosxSDKSpecificationsPath=$macosSdkPlatformPath/Developer/Library/Xcode/Specificati
ons
# packageTypesForMacOSXPath="$macosxSDKSpecificationsPath/MacOSX Package Types.xcspec"
# productTypesForMacOSXPath="$macosxSDKSpecificationsPath/MacOSX Product Types.xcspec"
macosxSDKSpecificationsPath=$macosSdkPlatformPath/Developer/Library/Xcode/PrivatePlugIn
s
packageTypesForMacOSXPath="$macosxSDKSpecificationsPath/IDEOSXSupportCore.ideplugin/Con
tents/Resources/MacOSX Package Types.xcspec"
productTypesForMacOSXPath="$macosxSDKSpecificationsPath/IDEOSXSupportCore.ideplugin/Con
tents/Resources/MacOSX Product Types.xcspec"
```

最后重新运行：

```
sudo bash md-install
```

即可

File /xxx/IDEiOSSupportCore.ideplugin/xxx/Embedded-Device.xcspec not found

- 问题：

Xcode 14.3.1的Mac中，报错：

```
➜  bin sudo bash md-install
...
File /Applications/Xcode.app/Contents/PlugIns/IDEiOSSupportCore.ideplugin/Contents/Reso
urces/Embedded-Device.xcspec not found
```

- 原因：`Xcode 13+` 之后，部分路径变化了，所以找不到对应路径
- 解决办法：从Xcode中搜索到Embedded-Device.xcspec的实际位置，然后拷贝到报错的路径（如果不存在，先创建对应目录）即可
- 具体步骤

（1）找到Embedded-Device.xcspec

```
➜  ~ cd /Applications/Xcode.app/Contents
➜  Contents find . -name Embedded-Device.xcspec
./Developer/Library/Xcode/Plug-ins/XCBSpecifications.ideplugin/Contents/Resources/Embed
ded-Device.xcspec
```

找到：

- `/Applications/Xcode.app/Contents/Developer/Library/Xcode/Plug-`

`ins/XCBSpecifications.ideplugin/Contents/Resources/Embedded-Device.xcspec`

（2）拷贝到报错目录

先新建该目录

```
sudo mkdir -p /Applications/Xcode.app/Contents/PlugIns/IDEiOSSupportCore.ideplugin/Contents/Resources/
```

再去拷贝：

```
sudo cp /Applications/Xcode.app/Contents/Developer/Library/Xcode/Plug-ins/XCBSpecifications.ideplugin/Contents/Resources/Embedded-Device.xcspec /Applications/Xcode.app/Contents/PlugIns/IDEiOSSupportCore.ideplugin/Contents/Resources/
```

确认文件的确存在：

```
➜  PlugIns ll /Applications/Xcode.app/Contents/PlugIns/IDEiOSSupportCore.ideplugin/Contents/Resources/
total 8
-rw-r--r--@ 1 root  wheel   437B 10 12 15:34 Embedded-Device.xcspec
```

最后重新去操作：

`sudo bash md-install`

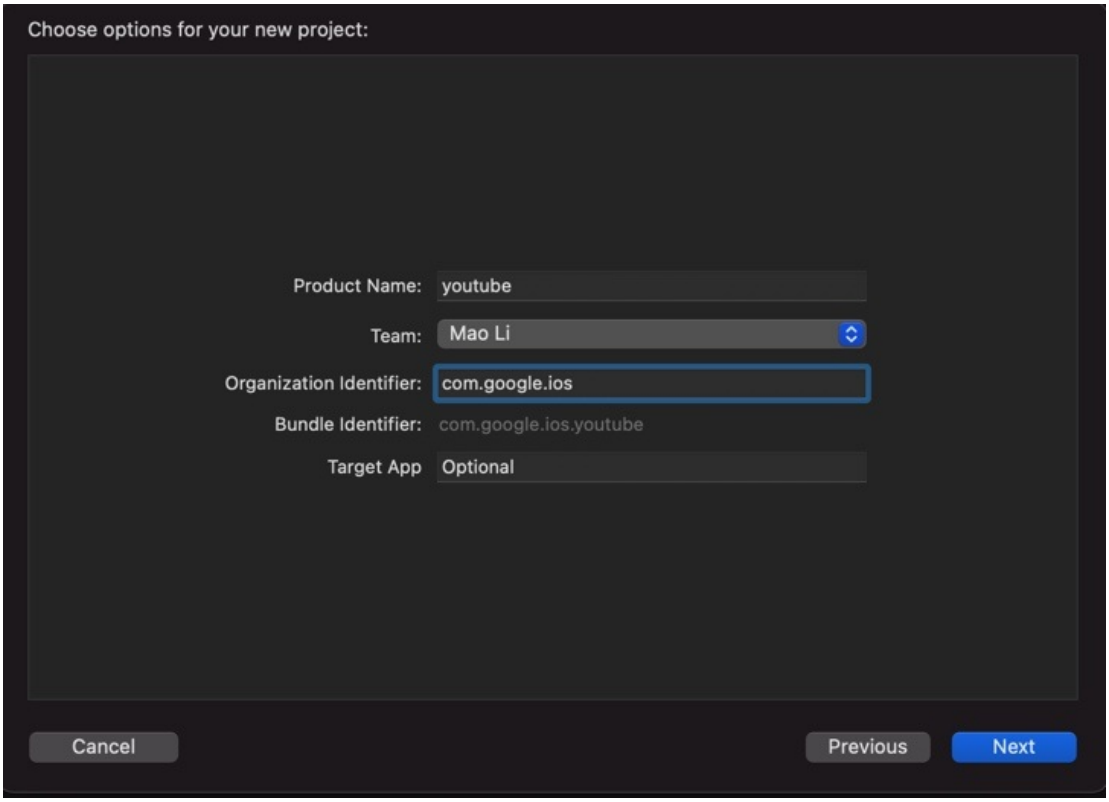
用MonkeyDev调试ipa

用Xcode+MonkeyDev去调试砸壳后的YouTube的ipa

- 概述
  - （1） `Xcode` 新建 `MonkeyDev` 的 `MonkeyApp` 项目
  - （2）设置基本参数
    - `Product` ： `youtube`
    - `Organization Identifier` ： `com.google.ios`
    - 自动生成包名： `com.google.ios.youtube`
      - 记得要和app真实包名是一致的
  - （3）右键 `TargetApp` -> `Add Files to youtube` ->选择YouTube的 `ipa`
    - 注意勾选：
      - `Destination` ： `Copy Items if needed`
        - 表示将ipa拷贝过来，而不是只是建立引用（链接）
      - `Added folders` ： `Create groups`
  - （4）确保已设置合适的目标部署iOS版本
    - 尽量让 `PROJECT` 和 `TARGETS` 中的iOS目标的版本一致
      - `PROJECT` -> `ProjectName` -> `Info` -> `Deployment Target` -> `iOS Deployment Target` ，比如设置为 `iOS 12.0`
      - `TARGETS` -> `ProjectName` -> `General` -> `Minimum Deployment` ，比如设置为 `iOS 12.0`
  - （5）确保 `Targets` 是 `youtube` （而**不是**youtubeDylib），点击▶按钮去启动调试，即可正常调试
  - 如果遇到各种问题
    - Unable to install
      - Could not inspect the application package
      - There was an internal API error
    - 可以：
      - 多试试几次
      - 或 `Xcode` -> `Clean Build Folder` ，一般均可解决问题
- 详解：

新建MonkeyDev项目

- Xcode中新建项目，选 `MonkeyDev` -> `MonkeyApp`

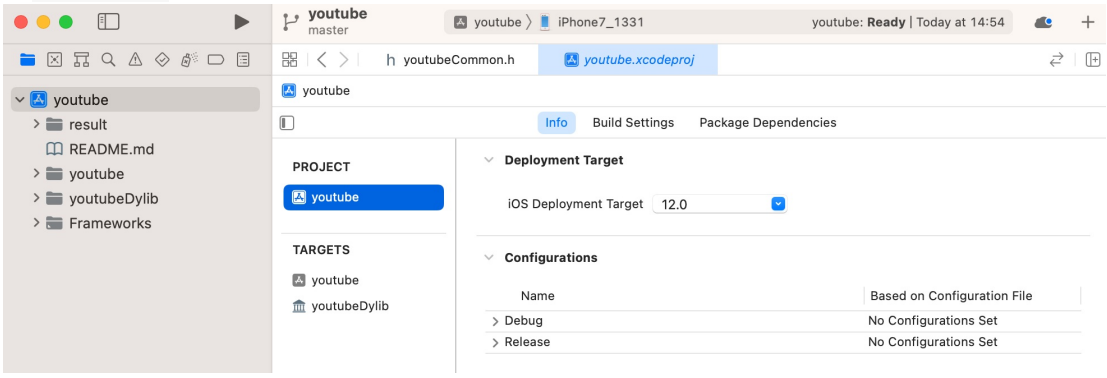
-
- 填写项目信息
  - 效果
    - 

  - 包名：`com.google.ios.youtube`
    - `Product Name`：`youtube`
    - `Organization Identifier`：`com.google.ios`
      - 自动生成包名：`com.google.ios.youtube`
    - `Target App`：`Optional`

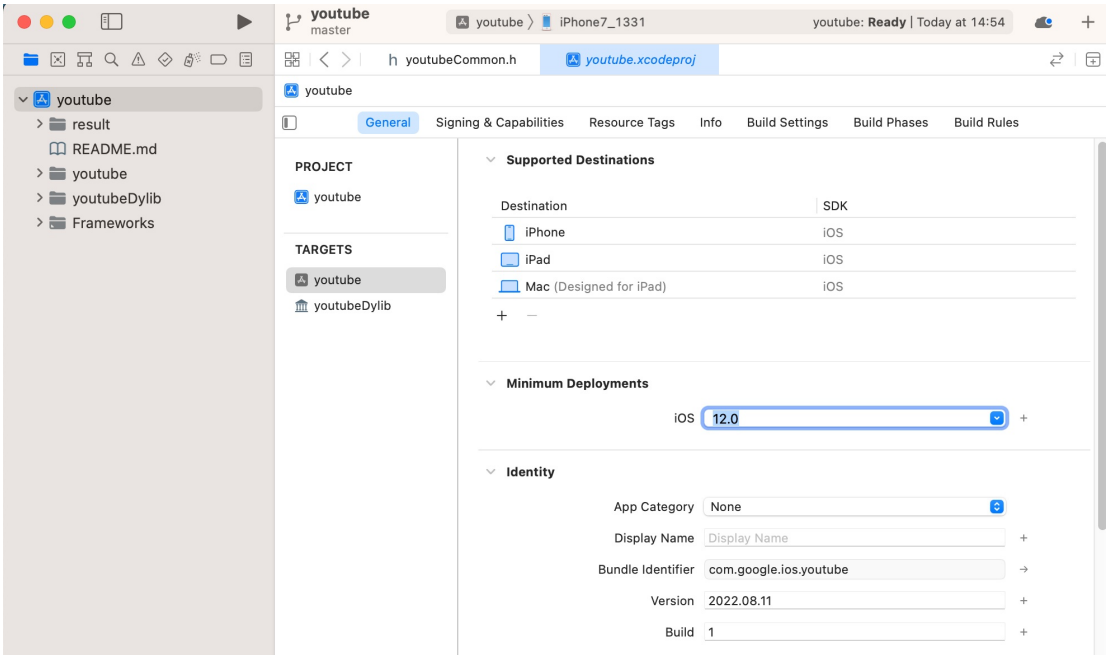
- 选择项目保存路径
  - 此处：`/Users/crifan/dev/DevRoot/YoutubeAdsFilter/Xcode/YouTube_1708`
- 新建好了`Xcode + MonkeyDev`的项目

确保已设置合适的目标部署iOS版本

- 尽量让`PROJECT 和 TARGETS`中的iOS目标的版本一致
  - `PROJECT -> ProjectName -> Info -> Deployment Target -> iOS Deployment Target`，比如设置为`iOS 12.0`

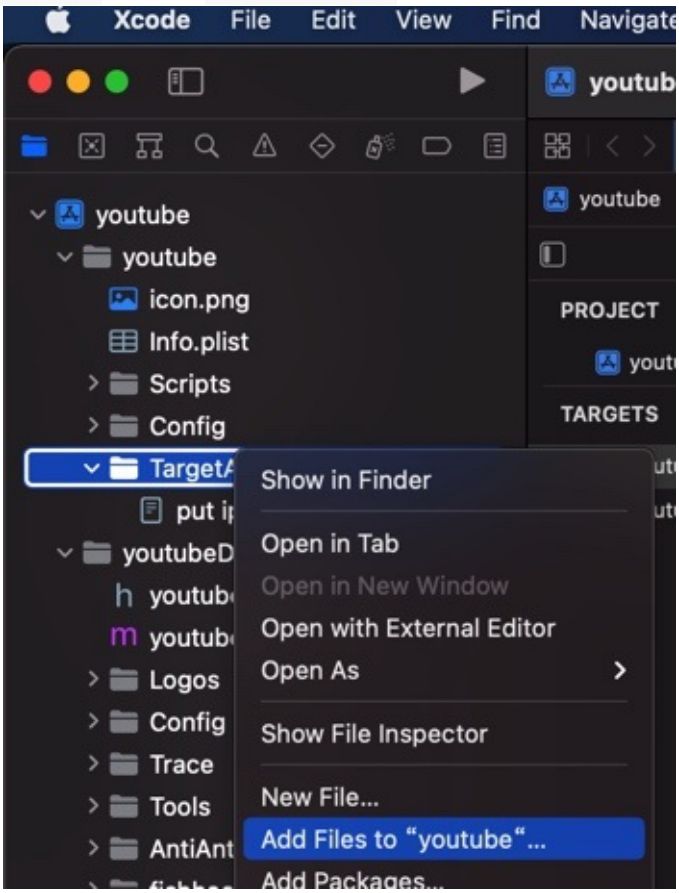


  - `TARGETS -> ProjectName -> General -> Minimum Deployment`，比如设置为`iOS 12.0`

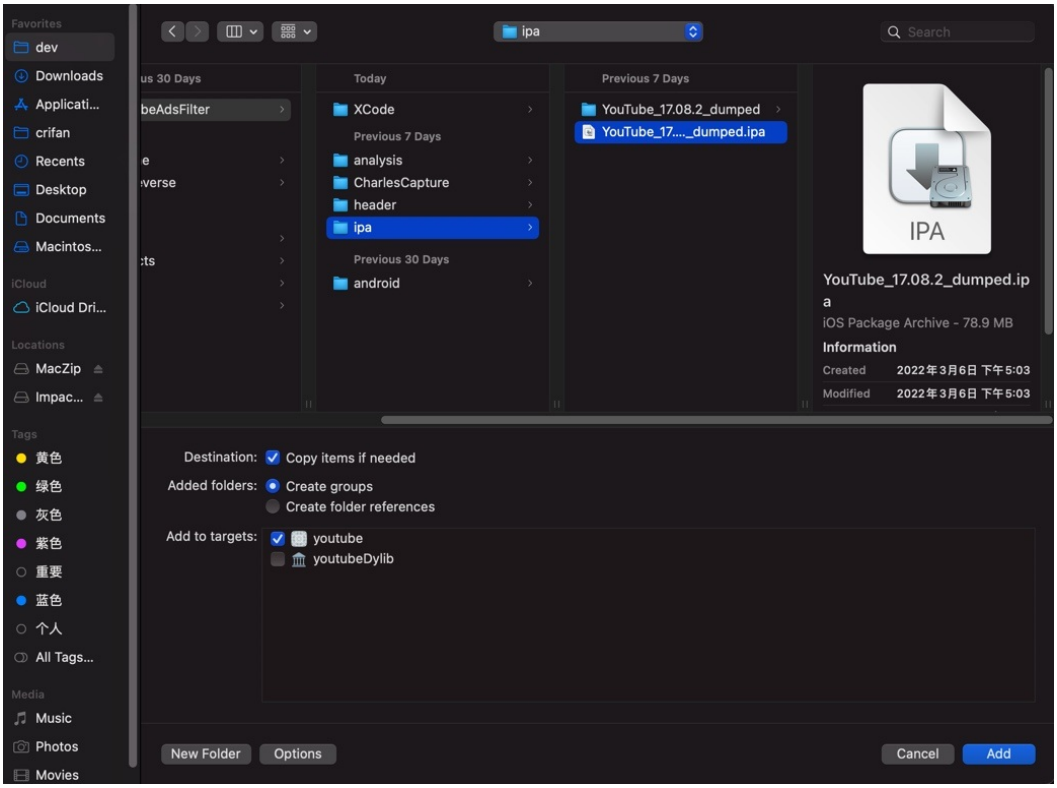


导入ipa

- 添加导入（砸壳后的）ipa
  - `TargetApp` ->右键-> `Add Files to`

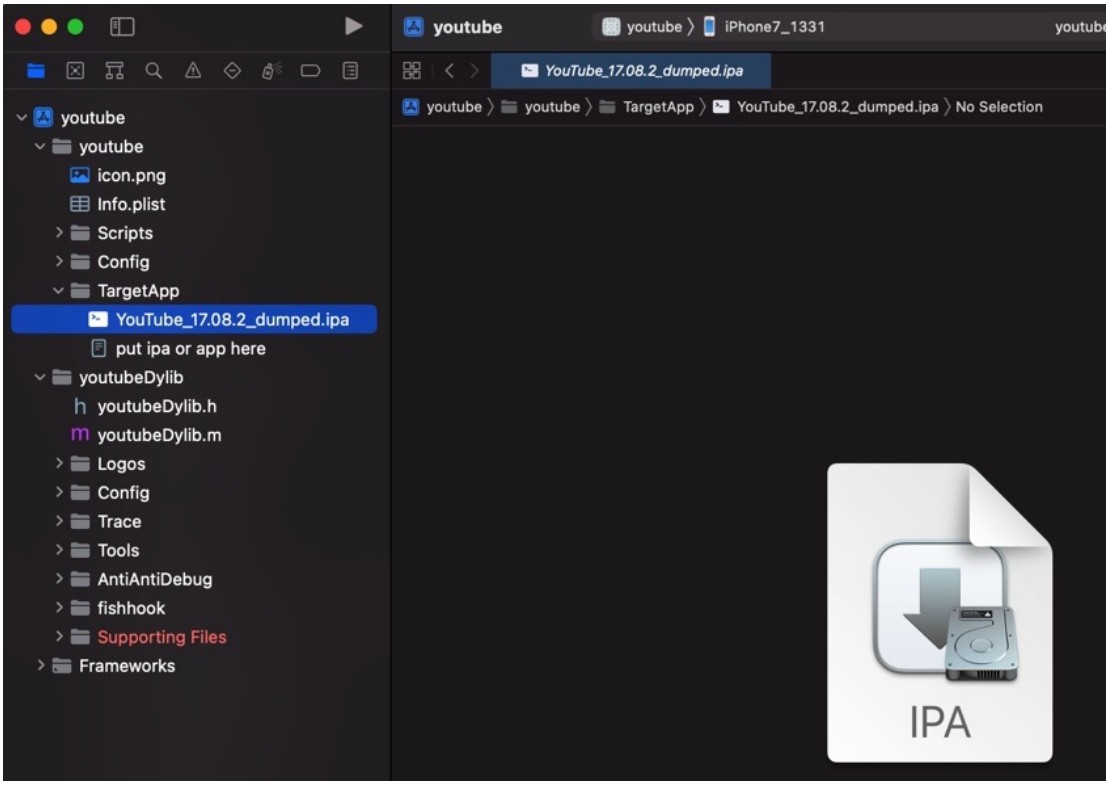


  - 选择ipa文件
    - 图



- 参数
  - `Destination` : `Copy Items if needed`
  - `Added folders` : `Create groups`

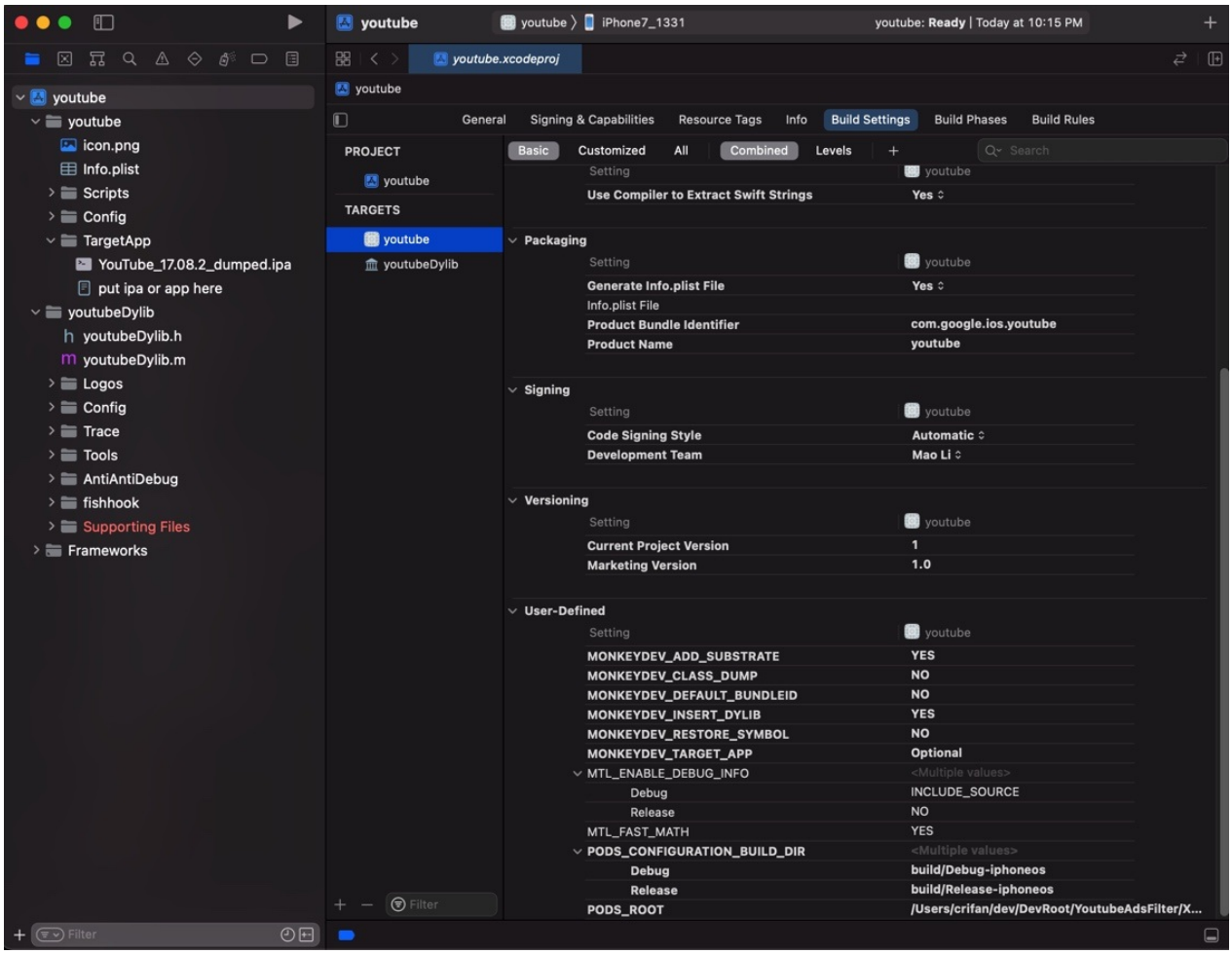
- 添加好了的ipa



确认（调整）MonkeyDev配置参数

注意：默认的 `TARGETS` 是 `youtubeDylib` ，要先去切换过去 `TARGETS` -> `youtube` ，才能看到配置。

去 `TARGETS` -> `youtube` 中确认此处MonkeyDev的配置参数（是你所希望的）：



此处参数配置值（多数是默认值）是：

- `MONKEYDEV_ADD_SUBSTRATE` = `YES`
- `MONKEYDEV_CLASS_DUMP` = `NO`
- `MONKEYDEV_DEFAULT_BUNDLEID` = `NO`
- `MONKEYDEV_INSERT_DYLIB` = `YES`
- `MONKEYDEV_RESTORE_SYMBOL` = `NO`
- `MONKEYDEV_TARGET_APP` = `Optional`

开始调试ipa

注意：默认的 `TARGETS` 是 `youtubeDylib` ，要先去切换过去 `TARGETS` -> `youtube` ，才能正常运行，安装ipa，开始调试。

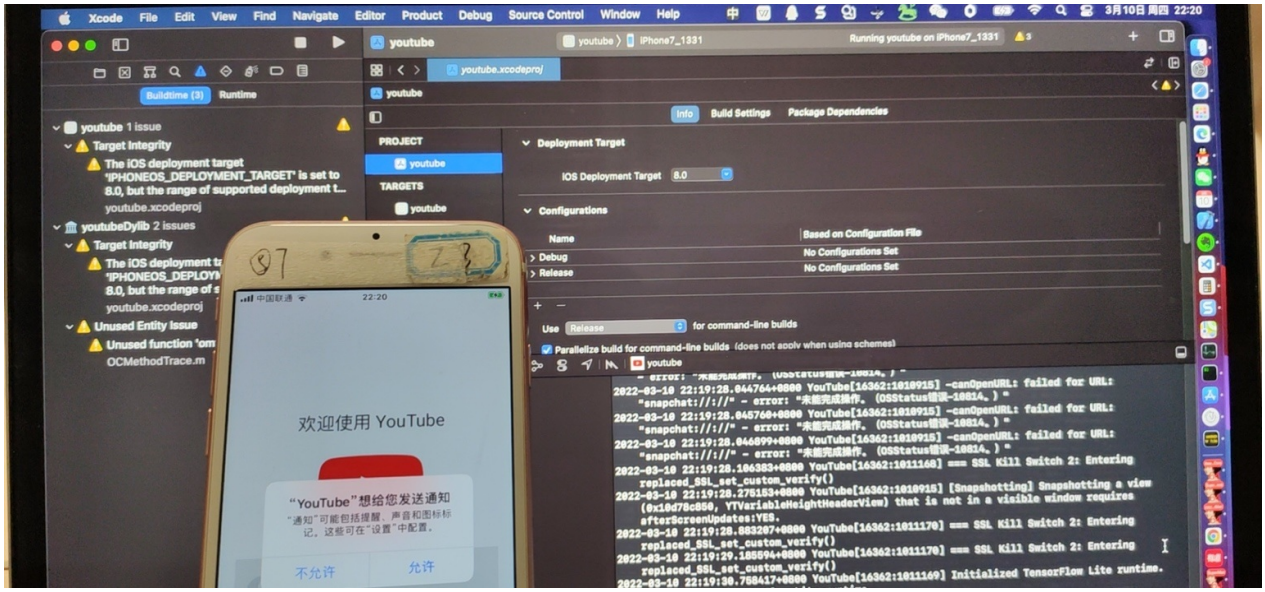
然后Xcode中即可去调试运行ipa：

- Building

- Installing

- Running

然后可以在 `iPhone` 真机上调试 `YouTube` 了：



常见问题

此处整理MonkeyDev调试编译运行ipa期间的常见问题。

ld: file not found: /usr/lib/libstdc++.dylib

- 问题

MonkeyDev编译链接时报错：

```
ld: file not found: /usr/lib/libstdc++.dylib
```

- 原因：`Xcode 10+` 之后=新版XCode，没了 `/usr/lib/libstdc++.dylib`
- 解决办法：网上找到缺失的 `/usr/lib/libstdc++.dylib` ，再安装拷贝到对应目录即可。
- 具体步骤

网上有人弄了个仓库，专门干这事。所以去下载代码和运行对应脚本即可。

```
git clone https://github.com/devdawei/libstdc-.git

cd libstdc-

chmod +x install-xcode_11+.sh

./install-xcode_11+.sh
```

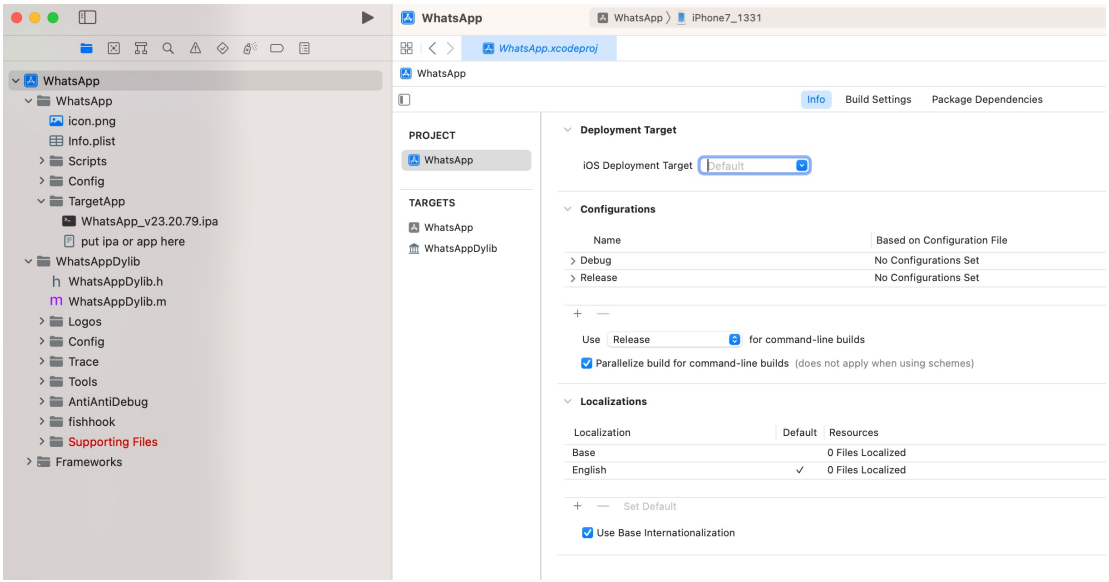
File not found: /xxx/arc/libarclite_iphoneos.a

- 问题

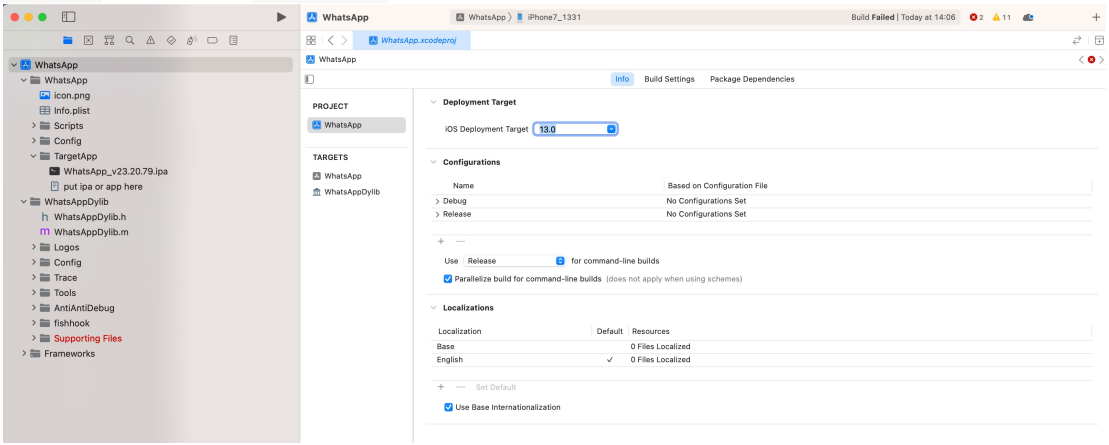
MonkeyDev调试ipa报错：

```
File not found: /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xcto
olchain/usr/lib/arc/libarclite_iphoneos.a
```

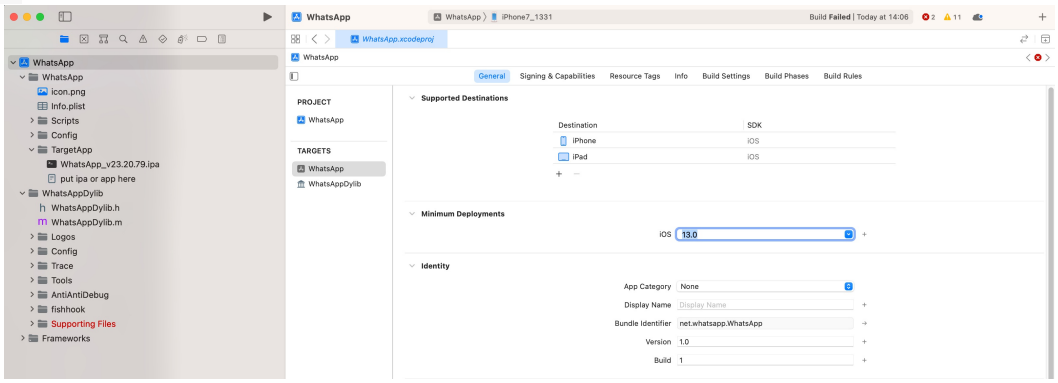
- 原因：没有设置PROJECT中的iOS部署目标的iOS版本
  - 导致是默认的值 `Default` =空值



- 解决办法：去设置PROJECT中的iOS部署目标的iOS版本
- 具体操作
  - 把项目的 `PROJECT` -> `Info` -> `Deployment Target` -> `iOS Deployment Target` 从默认的 `Default` =空值，改为 `iOS 13.0`



  - 注意：尽量保持和 `Targets` ->ProjectName-> `General` -> `Minimum Deployment` 中的 `iOS 13.0` 的值一致



自身包含

TODO：

- 要加上其他的?
  - AntiAntiDebug ?
  - trace?

---

class-dump

TODO：

- 【记录】支持iOS的Swift和ObjC混编的class-dump
- 【已解决】MonkeyDev安装失败：Failed to download AloneMonkey/frida-ios-dump/3.x/dump.py
- 【已解决】Mac中用class-dump导出YouTube头文件

- `class-dump`：是编译好的二进制支持swift混淆的版本
  - 对应路径：`/opt/MonkeyDev/bin/class-dump`
  - 版本信息

```
➜  ~ class-dump --version
class-dump 3.5 (64 bit) (Debug version compiled Sep 17 2017 16:24:48) compiled
Sep 17 2017 16:24:48
```

让MonkeyDev的class-dump全局可用

此次，之前已安装好 `iOSOpenDev` 的环境和设置了相关的环境变量：

- `~/.zshrc`

```
export iOSOpenDevPath=/opt/iOSOpenDev
export iOSOpenDevDevice=
export PATH /opt/iOSOpenDev/bin:$PATH
```

使得此处找到的 `class-dump` 是 `iOSOpenDev` 版本的：

```
➜  ~ which class-dump
/opt/iOSOpenDev/bin/class-dump
```

此处想要，把全局的，命令行行中找到的 `class-dump` 换成（支持Swift和ObjC混淆的） `MonkeyDev` 的

可以去：设置PATH环境变量，加上MonkeyDev的路径

编辑 `~/.zshrc` ，在最末尾加上：

```
export MonkeyDevPath /opt/MonkeyDev
export MonkeyDevDeviceIP=
export PATH /opt/MonkeyDev/bin:$PATH
```

保存退出。重启终端，即可实现我们的效果：

```
➜  ~ which class-dump
/opt/MonkeyDev/bin/class-dump
```

LLDBTools

心得

TODO：

- 【未解决】Mac中安装和搭建MonkeyDev+XCode的开发环境
- 【已解决】MonkeyDev安装失败：Failed to download AloneMonkey/frida-ios-dump/3.x/dump.py
- 【已解决】MonkeyDev安装报错：tar Error Failed to extract md-install file.tar.gz
- 【已解决】MonkeyDev的XCode项目编译报错：codesign_allocate error failed with exit code 34304 errno No such file or directory
- 【已解决】MonkeyDev的XCode编译：始终弹框安装codesign_allocate命令行工具
- 【已解决】XCode启动崩溃：Failed to register spec from DEiOSSupportCore.ideplugin couldn't register specification malformed property list dictionary required key Identifier not present
- 【已解决】MonkeyDev的XCode项目编译报错：Unable to install This application’s application-identifier entitlement does not match that of the installed application
-
- 【记录】用XCode和MonkeyDev调试Logos越狱插件代码的效果
- 【已解决】用XCode和MonkeyDev去调试iOS抖音app
- 【未解决】给MonkeyDev的pack.sh加上echo的log日志调试分析运行逻辑
- 【记录】分析XCode+MonkeyDev编译抖音ipa详细过程的log
- 【未解决】XCode+MonkeyDev调试iOS的ipa除了首次外后续调试均会异常
- 【基本解决】Mac中用MonkeyDev+XCode去调试抖音脱壳ipa

- 每次调试
  - 先Clean再Build：绕过bug，否则导致调试ipa会崩溃
    - 详见：
      - 【已解决】XCode+MonkeyDev调试18.9.0抖音的崩溃问题：先Clean后再调试
- Xcode中，新增.xm文件的流程
  - 先新增.xm文件，再Build出.mm，再把.mm加到要编译的文件列表
    - 好像还要做一个什么映射还是关联？以便确保 自动从.xm生成.mm ？

内部脚本逻辑

TODO：

整理下面多个帖子

MonkeyDev内部有一套自己的脚本，执行对应的预处理、编译、链接等等流程和逻辑。

下面介绍其中相关内容。

pack.sh

- 【未解决】XCode+MonkeyDev调试iOS的ipa除了首次外后续调试均会异常
- 【未解决】研究MonkeyDev的XCode中/opt/MonkeyDev/Tools/pack.sh脚本的内部逻辑
- 【未解决】给MonkeyDev的pack.sh加上echo的log日志调试分析运行逻辑
- 【记录】研究MonkeyDev中pack.sh中为何info.plist异常缺失图标等字段
-

md

- 【已解决】Xcode调试报错：/opt/MonkeyDev/bin/md No such file or directory

md-install

- 【已解决】Mac中MonkeyDev搭建环境运行md-install报错：File Xcode/Specifications/MacOSX Package Types.xcspec not found
- 【已解决】MonkeyDev安装报错：tar Error Failed to extract md-install file.tar.gz

项目代码结构

TODO：

- 【已解决】MonkeyDev的Xcode项目代码优化：新增独立文件youtubeCronet.xm
- 【已解决】MonkeyDev的Xcode项目代码优化：把公共部分提取到youtubeCommon.h
- 【已解决】MonkeyDev的Xcode项目代码优化：把hook代码移动到独立文件
- 【记录】优化MonkeyDev的YouTube代码：把Error部分提取到单独文件

待改进的细节

MonkeyDev调试时，偶尔有些细节，不是我们期望的=不尽如人意 的地方，整理如下：

image list的输出的加载镜像列表，其中app自身的路径，不是iPhone端的app的自身路径

概述：

```
(lldb) image list -o -f
[  0] 0x0000000002bfc000 /Users/crifan/Library/Developer/Xcode/DerivedData/WhatsApp-fuk
xiohktyjtjqfvzmmrwluorwjn/Build/Products/Debug-iphoneos/WhatsApp.app/WhatsApp
[  1] 0x00000001069fc000 /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/13.3.1
(17D50)/Symbols/usr/lib/dyld
...
```

- 其中的app的路径是
  - `/Users/crifan/Library/Developer/Xcode/DerivedData/WhatsApp-fukxiohktyjtjqfvzmmrwluorwjn/Build/Products/Debug-iphoneos/WhatsApp.app/WhatsApp`
- 很明显是个Mac端的app的路径
- 而不是移动端=iPhone端的app的实际路径
- 而我们期望的是：iPhone端的app的实际路径
- 其值应该是
  - 【记录】iOS逆向WhatsApp：lldb+debugserver调试时加载的image镜像列表
- 中

```
(lldb) image list -o -f
[  0] 0x0000000004c6c000 /private/var/containers/Bundle/Application/CCFD22D2-32EE-4
F23-9C81-226663100D40/WhatsApp.app/WhatsApp(0x0000000104c6c000)
[  1] 0x0000000108a44000 /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/13
.3.1 (17D50)/Symbols/usr/lib/dyld
...
```

- 的
  - `/private/var/containers/Bundle/Application/CCFD22D2-32EE-4F23-9C81-226663100D40/WhatsApp.app/WhatsApp`
- 这种，app在iPhone中实际的真实的路径

详见

- 【记录】iOS逆向WhatsApp：MonkeyDev调试时加载的image镜像列表

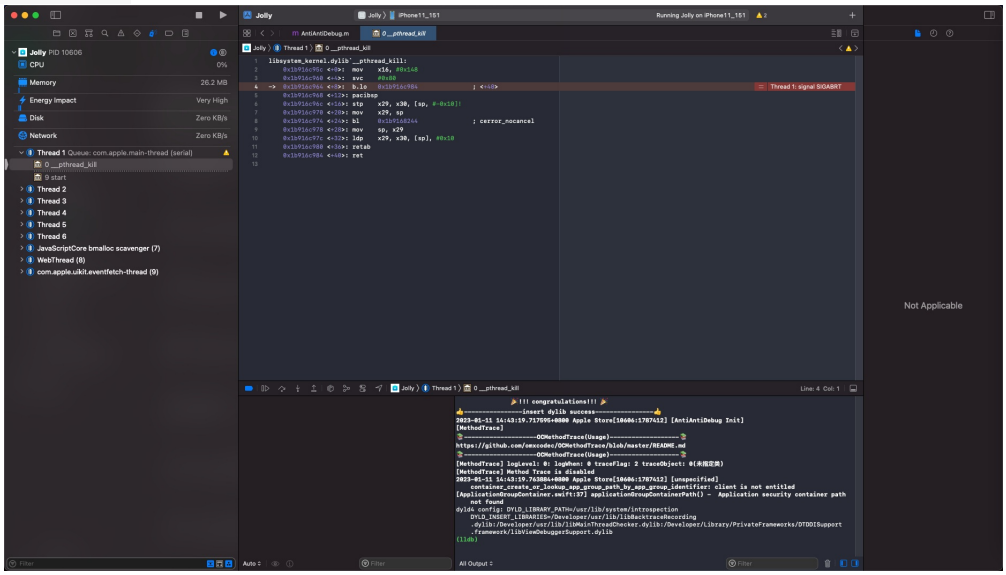
调试时各种崩溃和异常

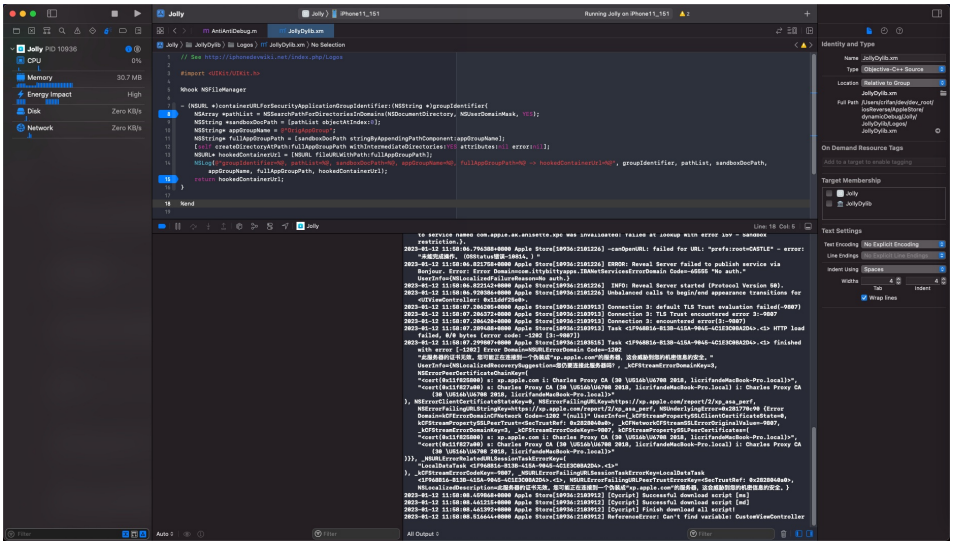
TODO：

- 【未解决】XCode+MonkeyDev调试iOS的ipa除了首次外后续调试均会异常
- 【未解决】iOS逆向AppleStore：为何MonkeyDev调试安装ipa后运行会出现各种出错

用MonkeyDev调试ipa期间，经常会遇到：各种的崩溃和异常

现象

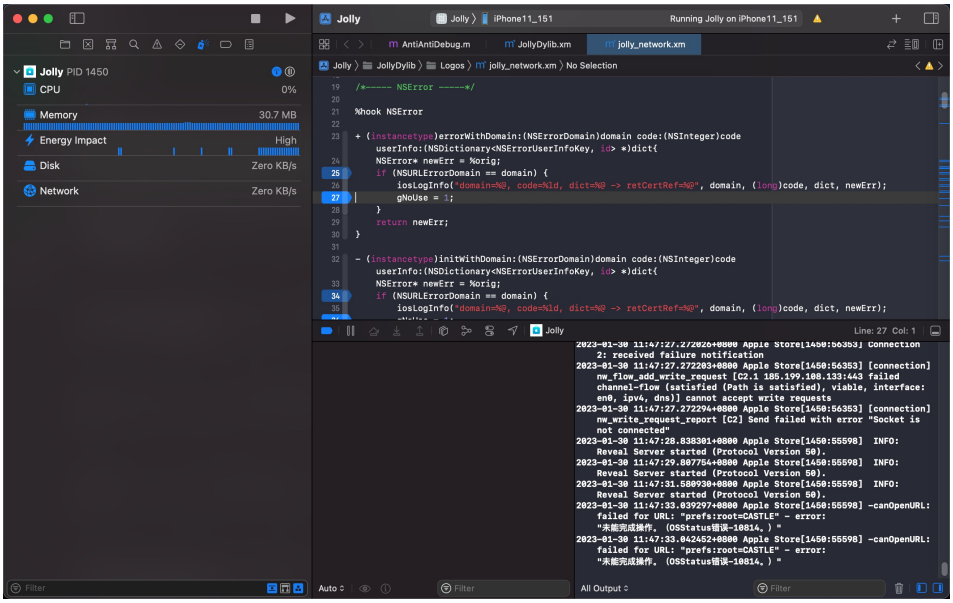
- MonkeyDev调试ipa的各种崩溃和异常
  - AppleStore
    - app group path问题
      - `[unspecified] container_create_or_lookup_app_group_path_by_app_group_identifier: client is not entitled`
      - 
    - Charles抓包证书出错问题 = 无法抓包，会报证书问题
      - 举例
        - 【未解决】MonkeyDev调试Apple Store报错：此服务器的证书无效。您可能正在连接到一个伪装成xp.apple.com的服务器，这会威胁到您的机密信息的安全

- 
- （从iCloud）同步Apple账户失败 = 账号登录问题：导致后续显示 `打开设置` 页面，让转去设置中去登录账号
  - 举例
    - 【未解决】iOS逆向AppleStore点击打开设置报错canOpenURL failed for URL prefs:root=CASTLE error 未能完成操作 OSStatus错误 -10814
      -

要在 Apple Store app 内购物，你需要有 Apple ID。

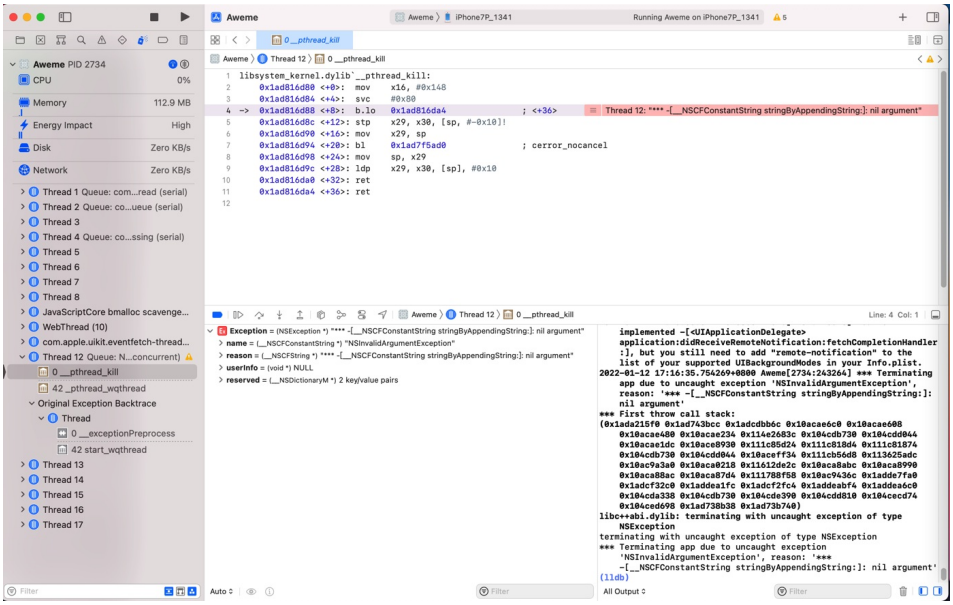
要使用有效的 Apple ID 登录，请前往你的 iOS 设备上的“设置”，然后轻点“登录 iPhone”。你也可以在这一步创建 Apple ID。

打开“设置”



- 【已解决】iOS逆向AppleStore：打开设置报错NSOSStatusErrorDomain Code -10814 _LSLine 225 _LSFunction _LSDOpenClient openURL
- 【未解决】iOS逆向AppleStore：无法自动登录Apple账号
- 【未解决】iOS逆向AppleStore：官网版本和砸壳版本对比找区别
- 【未解决】iOS逆向AppleStore：砸壳版本调试出现各种问题

- 抖音
  - NSString空字符串崩溃问题
    - 举例
      - 【规避解决】XCode的MonkeyDev调试抖音ipa崩溃：__NSCFConstantString stringByAppendingString nil argument



- 【未解决】通过XCode给stringByAppendingString加断点调试寻找抖音崩溃原因
- 【已解决】尝试解决XCode的MonkeyDev抖音ipa调试崩溃：hook函数stringByAppendingString

  - 等等

原因

- 根本原因
  - 概述：entitlement权限丢失
  - 细节
    - MonkeyDev调试ipa期间，会重新打包，会丢失掉原先app内部的完整的entitlement权限
    - 然后只使用了默认的最最基本的entitlement权限
    - 导致原先app的内置的很多其他对于app运行期间极其重要的entitlement权限，就丢失了
    - 所以就会导致后续运行期间，出现各种：崩溃和异常

底层技术细节

比如用MonkeyDev去调试 `Apple Store` 的ipa来说：

Xcode的编译期间的log可以看出编译过程是：

```
/usr/bin/codesign --force --sign 846361C864F687841B120144B1F1D0770BCB0EE6 --entitlements
 /Users/crifan/Library/Developer/Xcode/DerivedData/Jolly-edtiyeefjwnsmtdjblcgpzxtpvnt/B
uild/Intermediates.noindex/Jolly.build/Debug-iphoneos/Jolly.build/Jolly.app.xcent --tim
estamp\ none --generate-entitlement-der /Users/crifan/Library/Developer/Xcode/DerivedDa
ta/Jolly-edtiyeefjwnsmtdjblcgpzxtpvnt/Build/Products/Debug-iphoneos/Jolly.app
```

其中用到的 `Jolly.app.xcent` ，是：

（不论是否开启 `CODE_SIGN_INJECT_BASE_ENTITLEMENTS` ，都会使用的，通过默认的entitlement的模板所生成的）

默认的，内容非常少的，entitlement模板内容：

- `/Users/crifan/Library/Developer/Xcode/DerivedData/Jolly-edtiyeefjwnsmtdjblcgpzxtpvnt/Build/Intermediates.noindex/Jolly.build/Debug-iphoneos/Jolly.build/DerivedSources/Entitlements.plist`

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/Propert
yList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>application-identifier</key>
    <string>3WRHBBSBW4.com.apple.store.Jolly</string>
    <key>com.apple.developer.team-identifier</key>
    <string>3WRHBBSBW4</string>
    <key>get-task-allow</key>
    <true/>
</dict>
</plist>
```

从而覆盖掉

- 原始的，内容非常全的entitlement内容 == app原始的entitlement内容

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/Propert
yList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>com.apple.watchlist.private</key>
    <true/>
    <key>com.apple.authkit.client.private</key>
    <true/>
    <key>com.apple.developer.associated-domains</key>
    <array>
        <string>applinks:www.apple.com</string>
        <string>applinks:www.apple.com.cn</string>
        <string>applinks:concierge.apple.com</string>
        <string>applinks:reserve-prime.apple.com</string>
        <string>applinks:reserve-gb.apple.com</string>
        <string>applinks:reserve-cn.apple.com</string>
        <string>webcredentials:www.apple.com</string>
    </array>
    <key>com.apple.private.tcc.allow</key>
    <array>
        <string>kTCCServiceMediaLibrary</string>
        <string>kTCCServiceAddressBook</string>
    </array>
    <key>com.apple.accounts.idms.fullaccess</key>
    <true/>
    <key>com.apple.developer.pass-type-identifiers</key>
    <array>
        <string>MT9US5E2G8.*</string>
    </array>
    <key>application-identifier</key>
    <string>W74U47NE8E.com.apple.store.Jolly</string>
    <key>com.apple.accounts.appleaccount.fullaccess</key>
    <true/>
    <key>com.apple.itunesstored.private</key>
    <true/>
    <key>com.apple.private.MobileGestalt.AllowedProtectedKeys</key>
    <array>
        <string>UniqueDeviceID</string>
        <string>SerialNumber</string>
        <string>IntegratedCircuitCardIdentifier</string>
        <string>InternationalMobileEquipmentIdentity</string>
        <string>InternationalMobileEquipmentIdentity2</string>
        <string>IntegratedCircuitCardIdentifier2</string>
    </array>
    <key>fairplay-client</key>
    <string>187241837</string>
    <key>com.apple.Contacts.database-allow</key>
    <true/>
    <key>com.apple.developer.siri</key>
    <true/>
    <key>com.apple.private.applemediaservices</key>
    <true/>
    <key>com.apple.ap.adservicesd.statusconditionservice</key>
    <true/>
    <key>com.apple.developer.usernotifications.time-sensitive</key>
    <true/>
```

```
    <key>com.apple.private.appstored</key>
    <array>
        <string>IAPHistory</string>
    </array>
    <key>com.apple.springboard.opensensitiveurl</key>
    <true/>
    <key>com.apple.developer.in-app-payments</key>
    <array>
        <string>com.apple.ASA_AOS</string>
        <string>com.apple.ASA_EPC</string>
        <string>com.apple.ASA_AOS_KRYPTON</string>
        <string>com.apple.ASA_EPC_KRYPTON</string>
        <string>com.apple.ASA-AOS-ALT</string>
    </array>
    <key>com.apple.security.application-groups</key>
    <array>
        <string>group.com.apple.store.Jolly</string>
    </array>
    <key>com.apple.security.exception.shared-preference.read-write</key>
    <array>
        <string>com.apple.AvatarUI.Staryu</string>
        <string>com.apple.animoji</string>
    </array>
    <key>com.apple.developer.associated-appclip-app-identifiers</key>
    <array>
        <string>W74U47NE8E.com.apple.store.Jolly.Clip</string>
    </array>
    <key>com.apple.proactive.PersonalizationPortrait.Topic.readOnly</key>
    <true/>
    <key>com.apple.private.ind.client</key>
    <true/>
    <key>com.apple.security.exception.mach-lookup.global-name</key>
    <array>
        <string>com.apple.AppleMediaServicesUIDynamicService</string>
        <string>com.apple.appstored.xpc</string>
        <string>com.apple.proactive.PersonalizationPortrait.Topic.readOnly</string>
        <string>com.apple.corefollowup.agent</string>
        <string>com.apple.ndoagent</string>
        <string>com.apple.ind.xpc</string>
    </array>
    <key>aps-environment</key>
    <string>production</string>
    <key>com.apple.developer.default-data-protection</key>
    <string>NSFileProtectionCompleteUntilFirstUserAuthentication</string>
    <key>com.apple.security.exception.shared-preference.read-only</key>
    <array>
        <string>com.apple.suggestions</string>
    </array>
    <key>com.apple.security.exception.files.absolute-path.read-only</key>
    <array>
        <string>/var/mobile/Library/Preferences/com.apple.suggestions.plist</string>
    </array>
    <key>com.apple.private.ndoagent</key>
    <true/>
    <key>com.apple.ap.adservicesd.statusconditionclient.allow_read</key>
    <true/>
```

```
    <key>com.apple.private.tcc.allow-or-regional-prompt</key>
    <array>
        <string>kTCCServiceAddressBook</string>
    </array>
    <key>com.apple.developer.team-identifier</key>
    <string>MT9US5E2G8</string>
    <key>com.apple.coretelephony.Identity.get</key>
    <true/>
    <key>com.apple.private.avatar.store</key>
    <true/>
    <key>com.apple.accounts.appleidauthentication.defaultaccess</key>
    <true/>
    <key>com.apple.features.all-access</key>
    <true/>
</dict>
</plist>
```

注，查看entitlement的方式：

```
crifan@licrifandeMacBook-Pro~/dev/dev_root/iosReverse/AppleStore/fromiPhone11/AppleStore_TrollStoreInstalledOk_inited/Bundle/46830BF1-0DBF-4EE2-8084-1C0404BD7555 codesign -d --entitlements - Apple\ Store.app
Executable=/Users/crifan/dev/dev_root/iosReverse/AppleStore/fromiPhone11/AppleStore_TrollStoreInstalledOk_inited/Bundle/46830BF1-0DBF-4EE2-8084-1C0404BD7555/Apple Store.app/Apple Store
...
```

或：

```
crifan@licrifandeMacBook-Pro~/dev/dev_root/iosReverse/AppleStore/dynamicDebug/Xcode/Jolly/Jolly/TargetApp ldid -e Apple\ Store.app/Apple\ Store > AppleStore_embeded_entitlements.plist
```

由此导致了：

后续app正常运行期间，由于丢失了所需要的各种的entitlement权限，而运行崩溃或异常

举例：

丢失了原有的app group的entitlement权限的设置：

```
    <key>com.apple.security.application-groups</key>
    <array>
        <string>group.com.apple.store.Jolly</string>
    </array>
```

而导致了后续的app group path的问题：

```
2023-01-11 14:43:19.763884+0800 Apple Store[10606:1787412] [unspecified] container_create_or_lookup_app_group_path_by_app_group_identifier: client is not entitled
[ApplicationGroupContainer.swift:37] applicationGroupContainerPath() -  Application security container path not found
```

解决办法

彻底解决

- 彻底解决：暂时无解
  - 之前尝试解决，但是无法解决
    - 【无法解决】iOS逆向app：更改配置尝试解决MonkeyDev调试安装ipa各种错误
    - 抖音 = Aweme
      - 【记录】研究XCode+MonkeyDev后续调试ipa但不签名codesign能否解决崩溃问题
      - 【未解决】XCode的MonkeyDev参考和学习ipa安装过程和机制生成安装后不崩溃的抖音ipa
      - 【记录】分析XCode+MonkeyDev编译抖音ipa详细过程的log
    - AppleStore = Jolly.app
      - 【基本解决】iOS逆向Xcode中codesign：Xcode参数CODE_SIGN_INJECT_BASE_ENTITLEMENTS
      - 【未解决】iOS逆向AppleStore：codesign通过额外参数--preserve-metadata实现保留entitlement
      - 【未解决】iOS逆向AppleStore：Xcode编译时codesign不传入--entitlements参数即不使用entitlement文件
      - 【未解决】iOS逆向AppleStore：Xcode编译时codesign时如何指定合适的entitlement权限文件
      - 【未解决】iOS逆向AppleStore：Xcode编译时禁用codesign代码签名
      - 【未解决】iOS逆向Xcode中codesign：寻找BaseEntitlements.plist来源
      - 【未解决】iOS逆向Xcode中codesign：研究DerivedSources/Entitlements.plist的来源
      - 【未解决】iOS逆向Xcode中codesign：研究xcbuild文件的编译过程细节
      - 【无法解决】iOS逆向Xcode中codesign：找.app.xcent文件内容来源自己更改或替换默认内容
      - 【未解决】iOS逆向AppleStore：Xcode编译时codesign给参数--entitlements指定自己的entitlement文件
      - 【基本解决】iOS逆向Xcode中codesign：搞懂DerivedSources/Entitlements.plist的内容的来源
      - 【未解决】iOS逆向AppleStore：Xcode编译时如何保留修改后的entitlement文件或重签名的app
      - 【未解决】iOS逆向AppleStore：研究Xcode编译过程找二进制中entitlement丢失的原因
      - 【未解决】iOS逆向Xcode中自己指定entitlement：禁用自动管理签名
      - 【未解决】iOS逆向Xcode的codesign：看看编译时各种环境变量是否有用的
      - 【未解决】研究MonkeyDev的XCode中/opt/MonkeyDev/Tools/pack.sh脚本的内部逻辑
      - 【未解决】iOS逆向AppleStore：自己单独运行命令设置完整的entitlement权限
      - 【已解决】XCode中查看Build Phases中Run Script的sh脚本的log输出
      - 【未解决】给MonkeyDev的pack.sh加上echo的log日志调试分析运行逻辑
      - 【未解决】iOS逆向AppleStore：Xcode的build期间如何在Sign之后执行自定义命令
      - 【未解决】iOS逆向AppleStore：导致异常版本中的二进制中丢失plist的entitlement等信息的原因
      - 【已解决】Xcode调试ipa或app：确保项目debug-ipa正常调试运行

规避办法workaround

- 规避办法：改用其他调试手段
  - 优先推荐：`Xcode+iOSOpenDev`
    - iOS逆向调试：Xcode+iOSOpenDev
  - 其次可以考虑：`debugserver+lldb`
    - iOS逆向调试：debugserver+lldb

附录

下面列出相关参考资料。

参考资料

- iOS逆向调试：Xcode+iOSOpenDev
- iOS逆向调试：debugserver+lldb
-
- 【整理】iOS越狱插件开发工具：MonkeyDev
- 【未解决】MonkeyDev调试AppleStore报错：container_create_or_lookup_app_group_path_by_app_group_identifier
- 【已解决】XCode+MonkeyDev动态调试YouTube的ipa
- 【已解决】用MonkeyDev和XCode去调试17.8.0的抖音ipa
- 【已解决】Mac中安装和搭建MonkeyDev+XCode的开发环境
- 【已解决】MonkeyDev初始化报错：File /Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/Library/Xcode/Specifications/MacOSX Package Types.xcspec not found
- 【已解决】M2的Mac中给Xcode安装MonkeyDev出错：File /Applications/Xcode.app/Contents/PlugIns/IDEiOSSupportCore.ideplugin/Contents/Resources/Embedded-Device.xcspec not found
- 【已解决】MonkeyDev的XCode编译报错：ld file not found /usr/lib/libstdc++.dylib
- 【已解决】MonkeyDev编译运行报错：File not found arc libarclite_iphoneos.a
- 开始使用
- 非越狱App集成
- iOSOpenDev修改版MonkeyDev
- iOS逆向：2、MonkeyDev -- 记录（2020.12.24更） - leonlincq - 博客园
-