目录

iOS逆向开发：动态调试

- 最新版本：`v0.9`
- 更新时间：`20230315`

简介

介绍iOS逆向中的动态调试，包括动态调试的概览；以及调试代码逻辑方面，包括调试工具的MonkeyDev、lldb+debugserver、Frida等；以及相关子领域，比如反调试和反反调试等；以及查看界面元素的工具，比如Reveal、Cycript、LLDBTools、chisel、FLEX等；且详细介绍了Cycript的使用和心得；最后给出一些经验心得。

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_dynamic_debug: iOS逆向开发：动态调试

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

离线下载阅读

- iOS逆向开发：动态调试 PDF
- iOS逆向开发：动态调试 ePub
- iOS逆向开发：动态调试 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

iOS逆向动态调试概览

iOS逆向，从 `是否要运行代码` 的角度来说，分：

- 不要运行代码的：静态分析
- 要运行代码的： `动态调试`

此文主要介绍 `动态调试` 的相关内容：

- 输入=前提：砸壳出的ipa文件（或已把ipa安装到iOS设备中）
- 主要涉及的内容=领域
  - 调试代码逻辑
    - 常见调试工具
      - 图形界面： `Xcode + MonkeyDev`
      - 命令行： `debugserver + lldb`
      - `Frida`
      - IDA
    - 涉及到的相关子领域
      - `反调试` 和 `反反调试`
      - Xcode调试心得
        - 【整理Book】Xcode开发：调试心得
      - Xcode内置调试器：LLDB
  - 调试界面元素
    - `Reveal`
    - `Cycript`
    - （MonkeyDev的） `LLDBTools`
    - `chisel`
    - `FLEX`

反调试和反反调试

TODO：

- 【整理】iOS反越狱相关：反调试 反反调试
- 【已解决】iOS反调试和反反调试：syscall的ptrace
- 【未解决】iOS反调试和反反调试：svc 0x80的syscall的ptrace
- 【已解决】Mac中lldb调试iOS的app抖音报错：Process exited with status 45

由于现在多数iOS的app，都做了 `反调试` 的防护，导致想要能顺利调试iOS的app之前，都要去解决： `反反调试` 。

所以此处就分别涉及到：

- 正向的： `反调试`
- 逆向的： `反反调试`

反调试

反反调试

调试代码逻辑

TODO：

- 【未解决】如何调试iPhone中iOS的app

---

iOS逆向中的动态调试，其中主要是关于，用各种调试工具去调试代码逻辑。

常用调试工具有：

- `MonkeyDev`
- `lldb+debugserver`
- `Frida`
- IDA
  - 概述：其实IDA更多的是用来静态分析代码逻辑，偶尔用来 `动态调试`

以及相关心得：独立子教程

- Xcode调试心得
  - 【整理Book】Xcode开发：调试心得
- LLDB调试心得
  - Xcode内置调试器：LLDB

MonkeyDev

详见独立子教程：

iOS逆向开发：MonkeyDev调试 (crifan.org)

lldb+debugserver

TODO：

- 【已解决】把增加了权限的debugserver拷贝到越狱iPhone中
- 【已解决】debugserver带日志运行报错：Failed to open log file for writing errno 1 Operation not permitted
- 【已解决】debugserver启动iOS的app抖音报错：Segmentation fault 11
- 【已解决】用debugserver和lldb去调试iOS的app

---

iOS逆向调试代码逻辑的调试工具之一是：命令行的 `lldb + debugserver`

- `debugserver`
  - 是什么：一个终端的应用
    - 也是：`XCode` 去调试iOS设备中程序时候的进程名
  - 在哪里：iOS设备中
    - 位置：`/Developer/usr/bin/debugserver`
    - 注：iOS中默认没安装
      - iOS中安装 `debugserver`
        - 在设备连接过一次 `Xcode` ，并在 `Window` -> `Devices` 中添加此设备后
          - `debugserver` 才会被 `Xcode` 安装到 `iOS` 的 `/Developer/usr/bin/` 下
  - 作用：作为服务端，接受来自远端的 `gdb` 或 `lldb` 的调试
    - 可以理解为：`lldb` 的 `server`
  - 为何需要
    - iOS中，由于App运行检测到越狱后会直接退出，所以需要通过 `debugserver` 来启动程序
  - 通过 `debugserver` 来启动程序
    - 举例
      - `debugserver -x backboard 0.0.0.0:1234 ./*`
      - `debugserver *:1234 -a "MoneyPlatListedVersion"`

从技术上，应属于：LLDB的远程调试，需要用到： `lldb-server`

- lldb-server 远程调试
  - 分2个端
    - `lldb client`
      - 运行在 local system

        - 比如 `Linux` 、 `Mac`
    - `lldb server`
        - 不同平台
            - `Linux` 和 `Android` ： `lldb-server`
                - 不依赖于 `lldb`
                    - 因为：已静态链接包含了 `LLDB` 的核心功能
                    - 对比： `lldb` 是默认是动态链接 `liblldb.so`
            - `Mac` 和 `iOS` ： `debugserver`
        - 运行在 remote system
        - 实现了remote-gdb的功能
    - 两者通讯
        - 用的是： `gdb-remote` 协议
            - 一般是在TCP/IP之上运行
- 细节详见：
    - `docs/lldb-gdb-remote.txt`
- 资料
    - 主页
        - Remote Debugging — The LLDB Debugger
            - http://lldb.llvm.org/use/remote.html

Frida

TODO：

- 【记录】iOS的iPhone中安装Frida
- 【未解决】用Frida动态调试iOS版抖音app
- 【已解决】frida启动抖音app报错：Failed to attach need Gadget to attach on jailed iOS
- 【已解决】Frida中如何通过frida启动被测app程序iOS版抖音
- 【未解决】frida去hook函数_dyld_get_image_name时打印参数为字符串
- 【未解决】frida调试抖音app去hook函数：_dyld_get_image_name
- 【已解决】frida调试进程直接运行不要每次都输入%resume才运行
- 【已解决】用frida启动hook调试iOS抖音app
- 【未解决】Mac中frida-trace报错：Failed to spawn unable to find process with name
- 【未解决】用Frida的frida-trace去hook函数iOS版抖音
- 【记录】frida的frida-ps用法
- 【记录】iOS的iPhone中安装Frida

- `Frida`
  - 主要用途：iOS逆向期间，写hook函数，动态调试和研究代码逻辑
  - 主页
    - https://www.frida.re
      - Frida • A world-class dynamic instrumentation framework
      - Inject JavaScript to explore native apps on Windows, macOS, GNU/Linux, iOS, Android, and QNX
      - you can make experiments without process restarting.
      - Of course production-level tweaks must be supplied as native .dylib/.plist pair.

调试界面元素

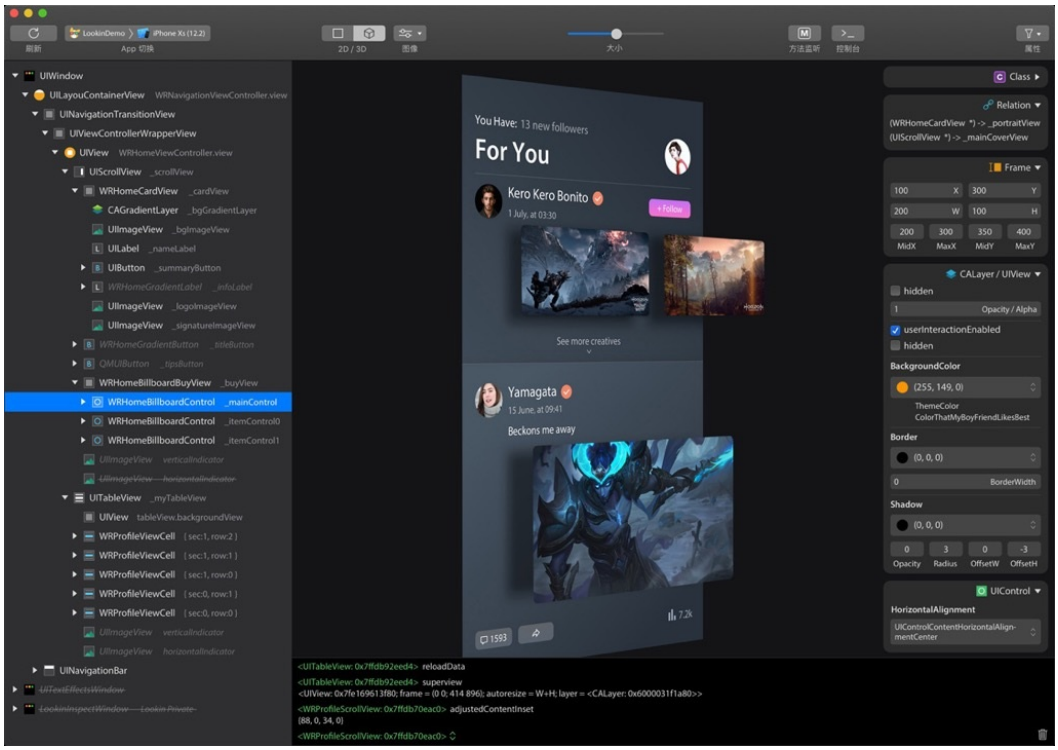
TODO：

- 【整理】页面元素调试结果对比：Reveal、Cycript、LLDBTools、chisel
- 【记录】XCode+MonkeyDev动态调试抖音：从点赞关注UI界面入手找底层代码逻辑

---

iOS逆向的动态调试，也常会，从app的界面入手找对应的按钮等元素，此时就会涉及到：调试界面元素

常用的iOS的app的界面调试工具：

- `Reveal`
- `Cycript`
- （MonkeyDev的）`LLDBTools`
- `chisel`
- `FLEX`
- 其他
  - LookinLoader
    - https://github.com/creantan/LookinLoader
    - 

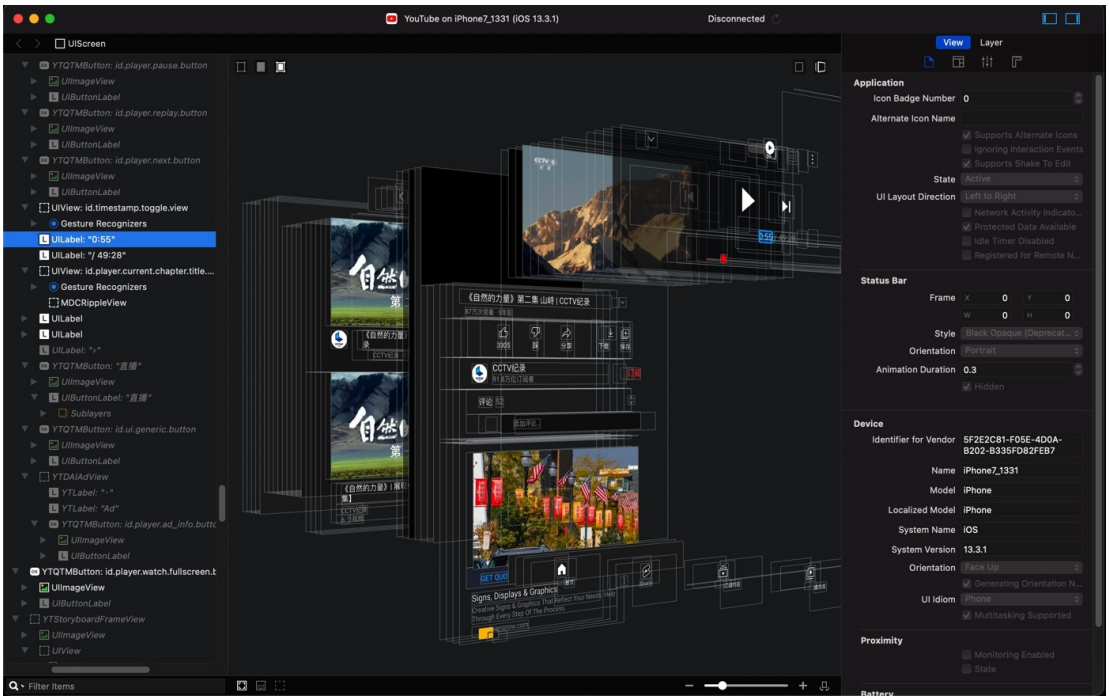
Reveal

TODO：

- 【已解决】用Reveal查看抖音UI界面中点赞关注按钮相关的类和实现
- 【记录】找抖音关注按钮响应事件：pactions
- 【记录】通过Reveal查看页面元素找YouTube广告相关类
- 【记录】通过Reveal查看YouTube广告页面元素

---

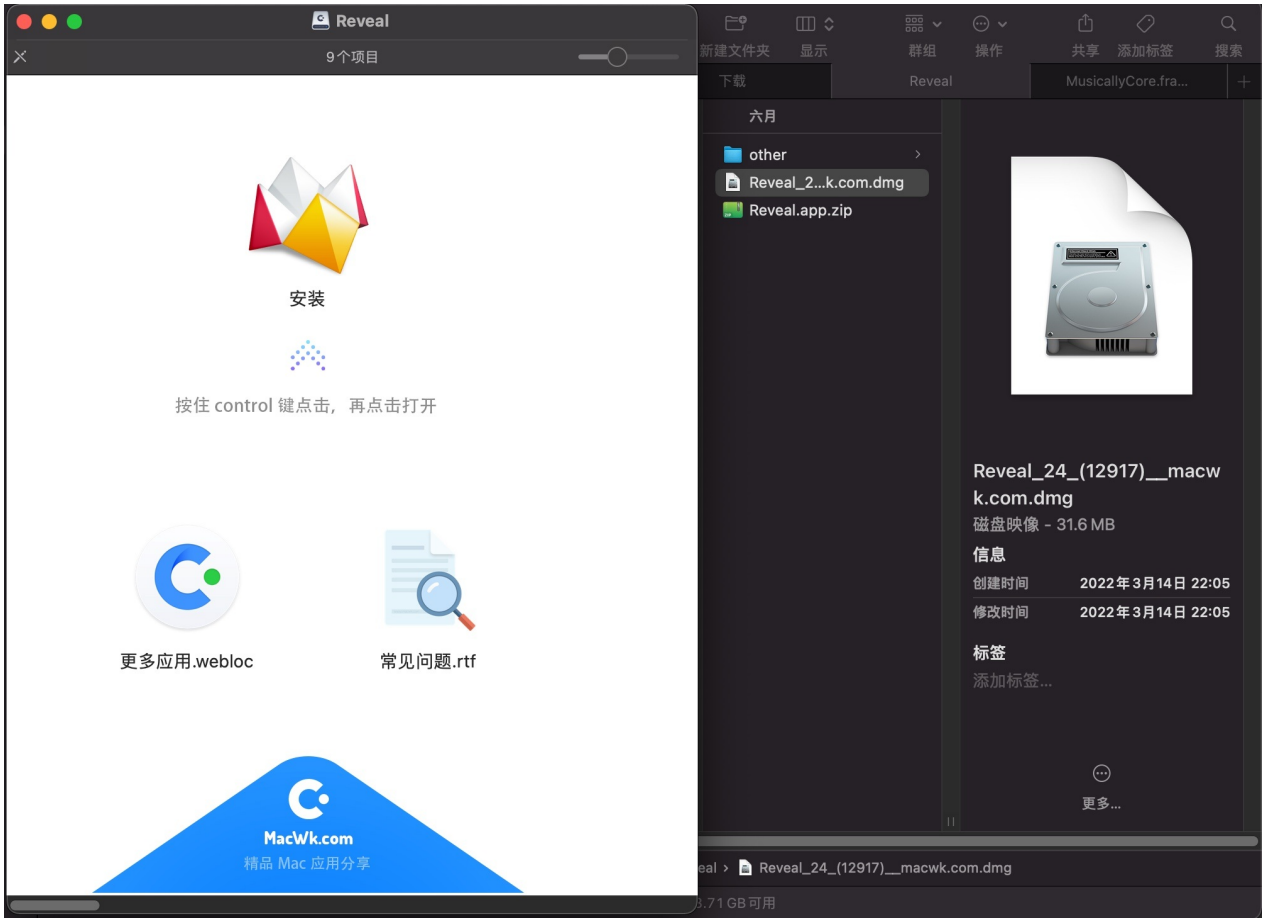
iOS逆向中，用来调试界面元素，比较好用的工具之一就是：`Reveal`

- 效果举例
  - YouTube
    - 

Xcode+MonkeyDev配合Reveal调试UI界面元素

下载安装Reveal

从这里下载到Reveal24(12917)__macwk.com.dmg__macwk.com.dmg)，然后安装：



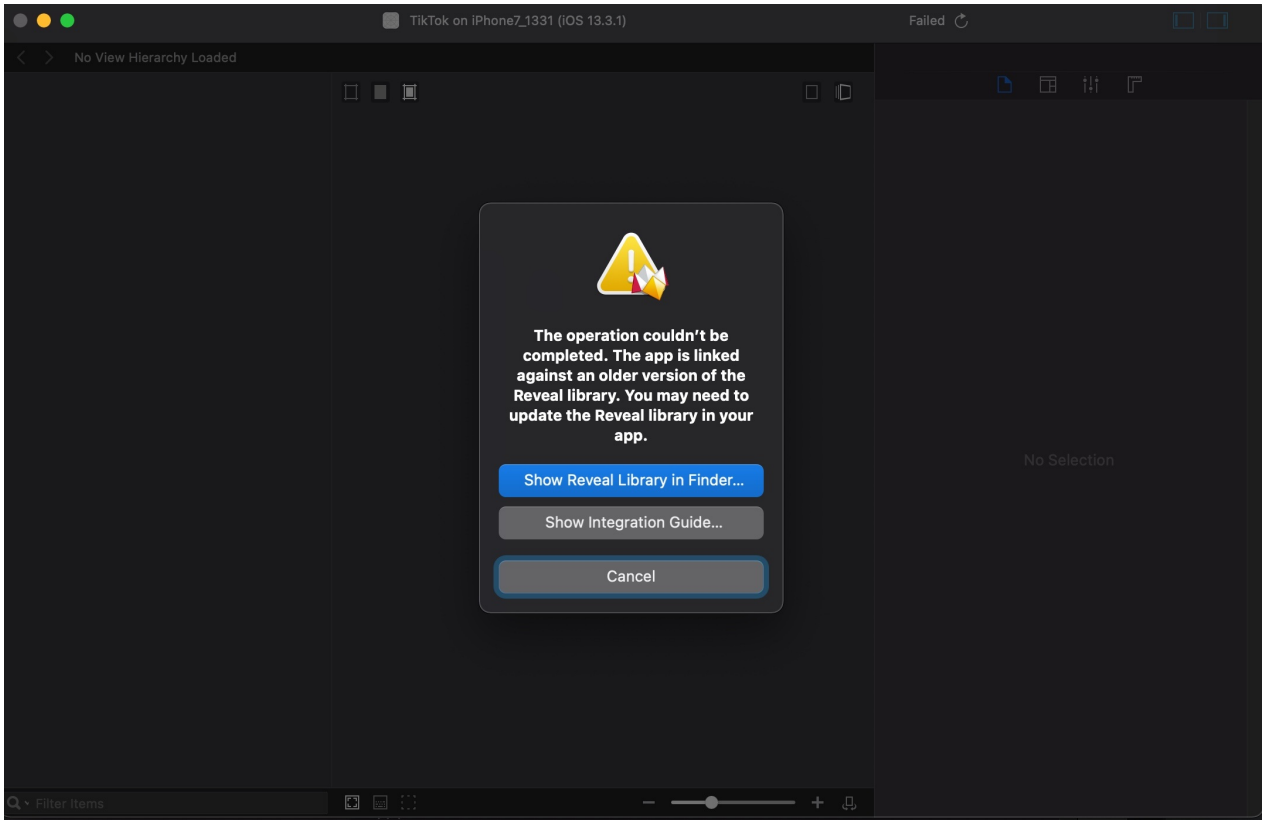
确保 `RevealServer.framework` 版本一致

- `Mac`：`Reveal.app` 中的 `RevealServer.framework`
  - 默认位置：`/Users/{YourUserName}/Library/Application Support/Reveal/RevealServer/iOS/RevealServer.framework`
- `iPhone`：所运行的是 `MonkeyDev` 内部集成的 `RevealServer.framework`
  - 默认位置：`/opt/MonkeyDev/Frameworks/RevealServer.framework`

要确保版本一致。

否则 `Reveal.app` 连接 `iPhone` 调试时会报错：

```
The operation couldn't be completed. The app is linked against an older version of the Reveal library. You may need to update the Reveal library in your app.
```



解决办法：

点击弹框中的：`Show Reveal Library in Finder…`，会自动打开（当前`Mac`中）最新版本的`RevealServer.framework`

`/Users/{YourUserName}/Library/Application Support/Reveal/RevealServer/iOS/RevealServer.framework`

然后将其拷贝过去，替换掉旧的`MonkeyDev`的：

`/opt/MonkeyDev/Frameworks/RevealServer.framework`

即可。

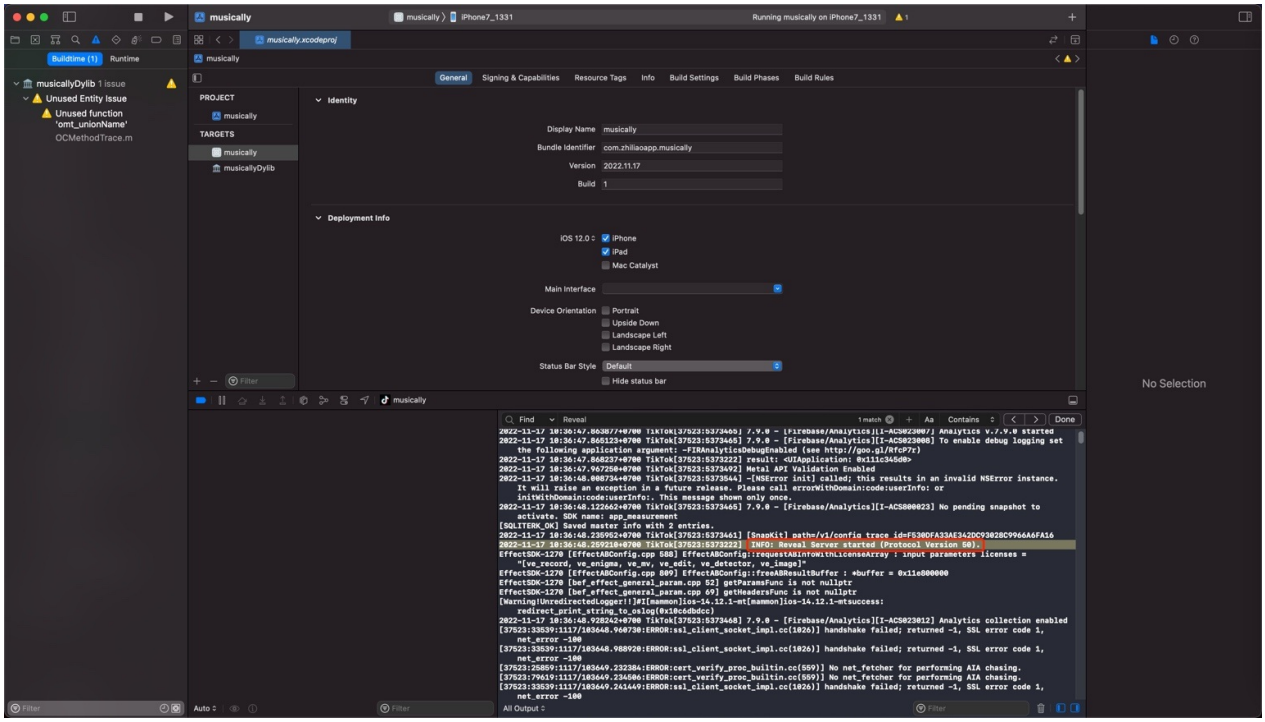
注：

- `/opt/MonkeyDev/Frameworks/`
  - 是MonkeyDev的常见的默认的安装路径
- 复制时需要root权限
  - 所以命令行复制时，需要sudo，否则会报错没有权限
  - Finder界面中复制时，需要输入当前Mac用户的密码

用`XCode`+`MonkeyDev`调试`iOS`的`app`(`ipa`)

其中Xcode中能输出=能搜到对应的log：

```
2022-11-17 10:36:48.259210+0700 TikTok[37523:5373222]  INFO: Reveal Server started (Protocol Version 50).
```



表示Reveal Server服务已启动

注意：

- 确保最后一条Reveal的log是Started
  - -》意思是Reveal的确在运行
  - 否则也可能遇到，中间Reveal是Started，但之后还有Stopped的log，则表示Reveal服务是停止掉了
  - 那样的话，Reveal是无法使用的

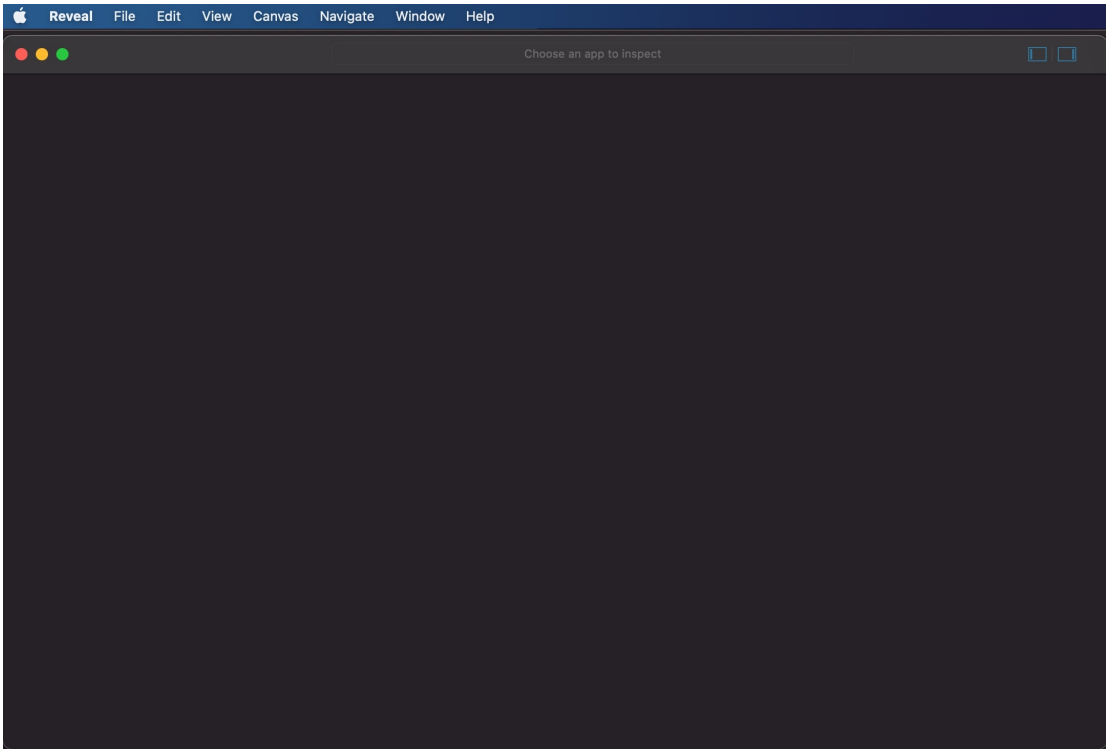
打开Mac中的 `Reveal.app` ，去连接和调试设备中的app的界面

`Reveal` -> `File` -> `New Tab`

点击 `Discovered` 所显示出iPhone设备了

注意：

- 首次启动Reveal后，往往看不到iPhone设备（中的app）
  - 图

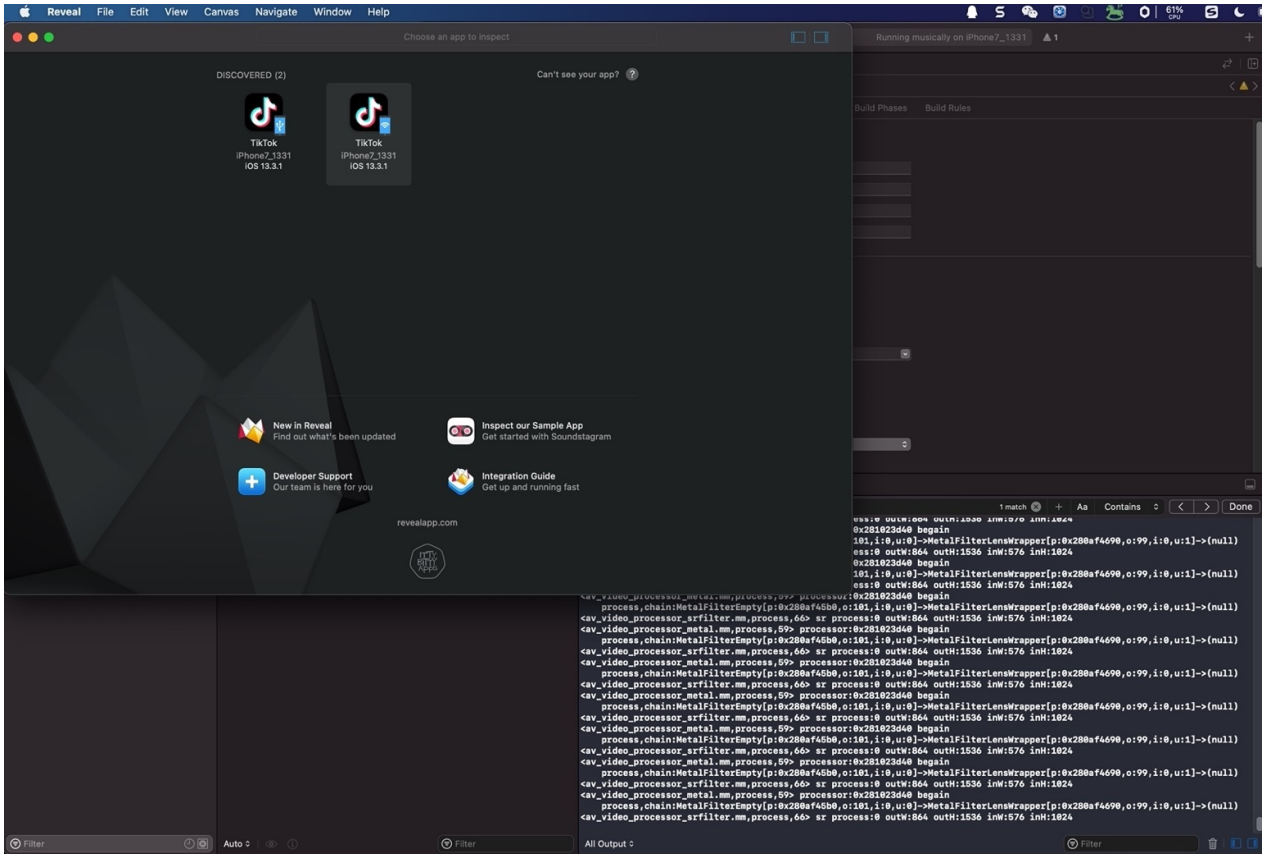


    - 解决办法：
        - 关闭Reveal，重启Reveal，即可。

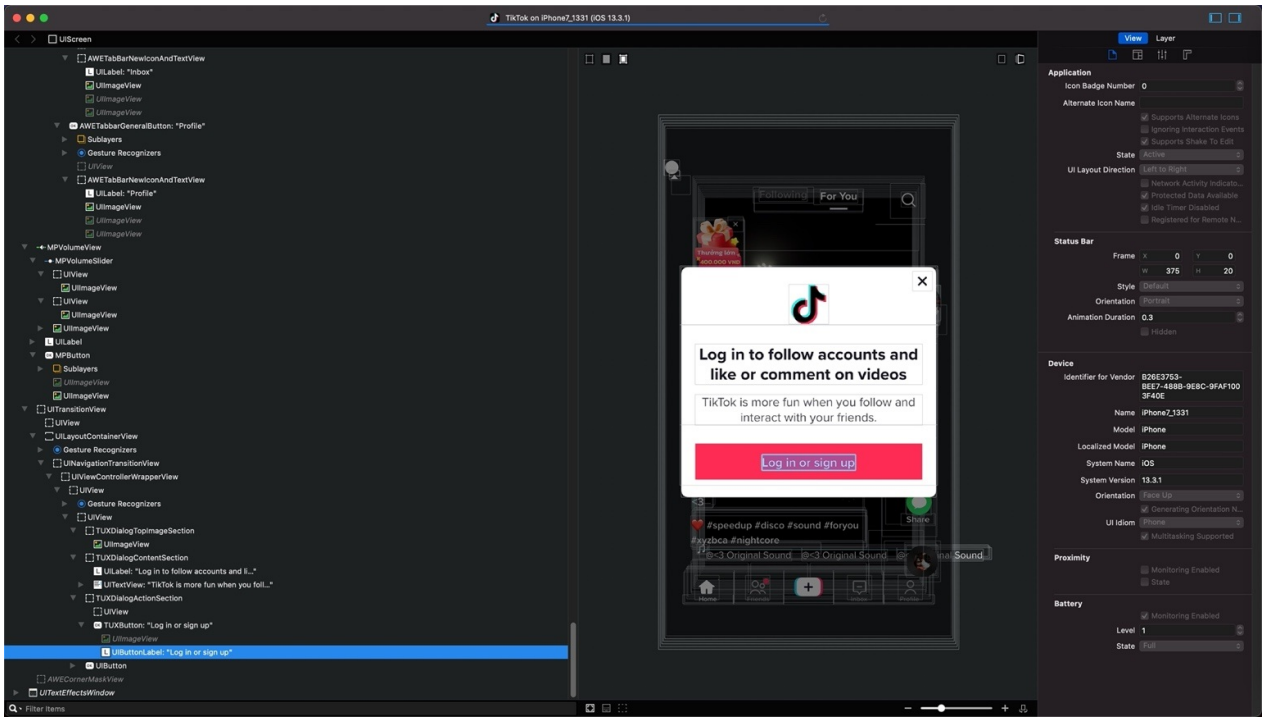
一般正常会出现2个按钮：

- Wifi
- USB

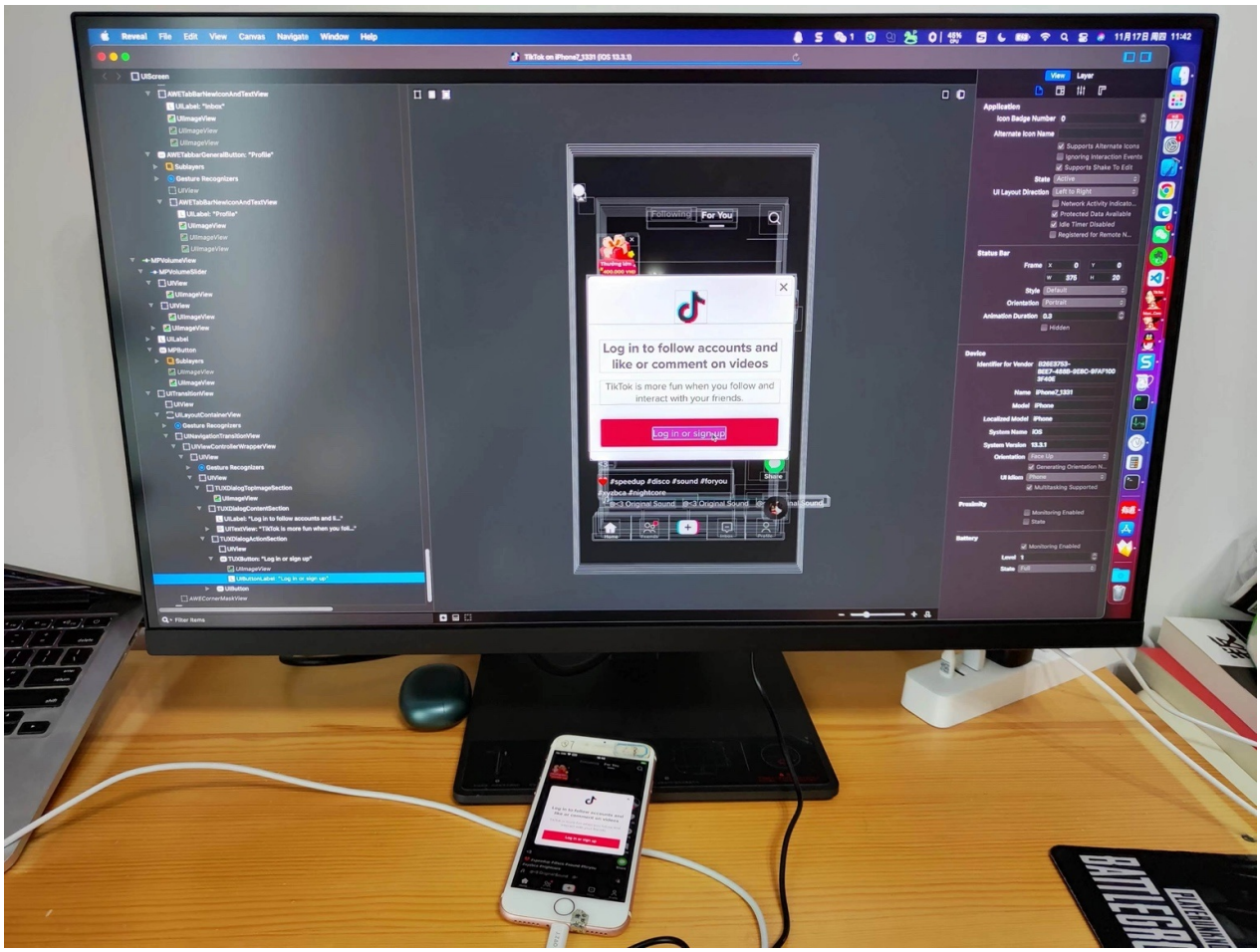
按道理USB的连接更稳定些，所以一般点击USB的



即可连接和正常调试app的UI界面元素了：

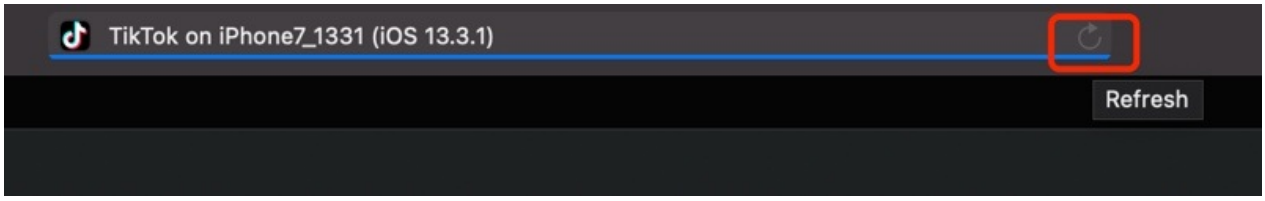


Mac电脑+iPhone手机的效果：



刷新页面

如果app端页面刷新了，可以点击Tab顶部的右上角的刷新按钮，即可刷新



注：

此处Refresh按钮是灰色的，原因是：此处Tiktok的app的UI界面元素内容太多，导致一直在加载，始终加载不能完全结束，所以无法刷新

不过一般无所谓，可以重新关掉窗口，重新点击连接设备，从而分析app上最新的界面元素的。

Cycript

TODO：

- 【部分解决】用Cycript查看抖音UI界面元素以寻找关注按钮所属元素
- 【已解决】用MonkeyDev中Cycript去调试YouTube的UI页面的元素

---

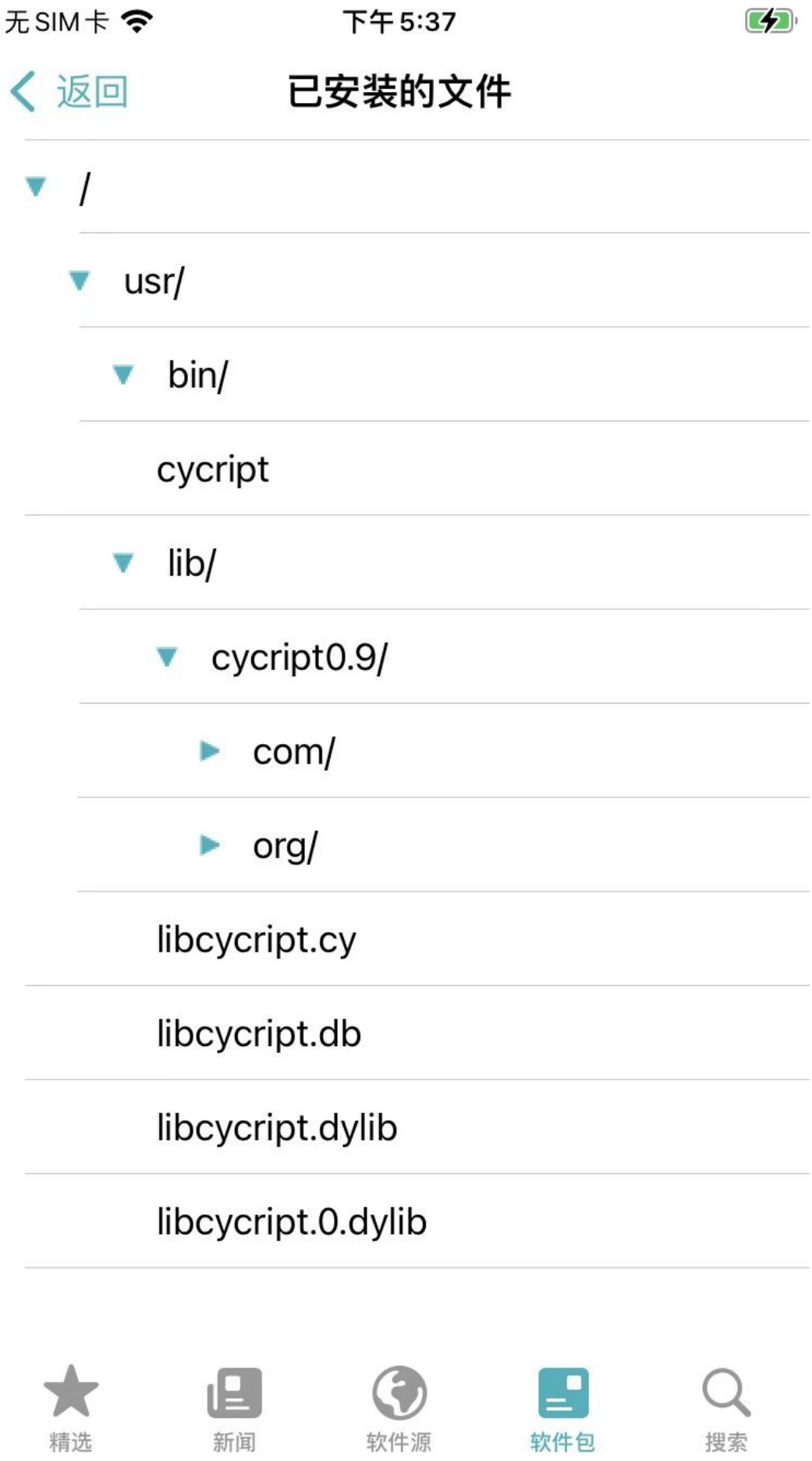
iOS逆向的调试界面元素的工具，也有命令行的：`Cycript`

- Cycript
  - 官网
    - http://www.cycript.org/
    - 文档
      - http://www.cycript.org/manual/
  - 有用资料
    - Cycript Tricks - iPhone Development Wiki

Cycript初始化环境

- Cycript环境搭建=初始化
  - `Sileo` / `Cydia` 中安装 `cycript` 插件即可
    - 步骤
      - 如果没有，需要先添加软件源：
        - https://strap.palera.in/
      - 然后去搜索：`cycript`，并安装，即可
    - 效果

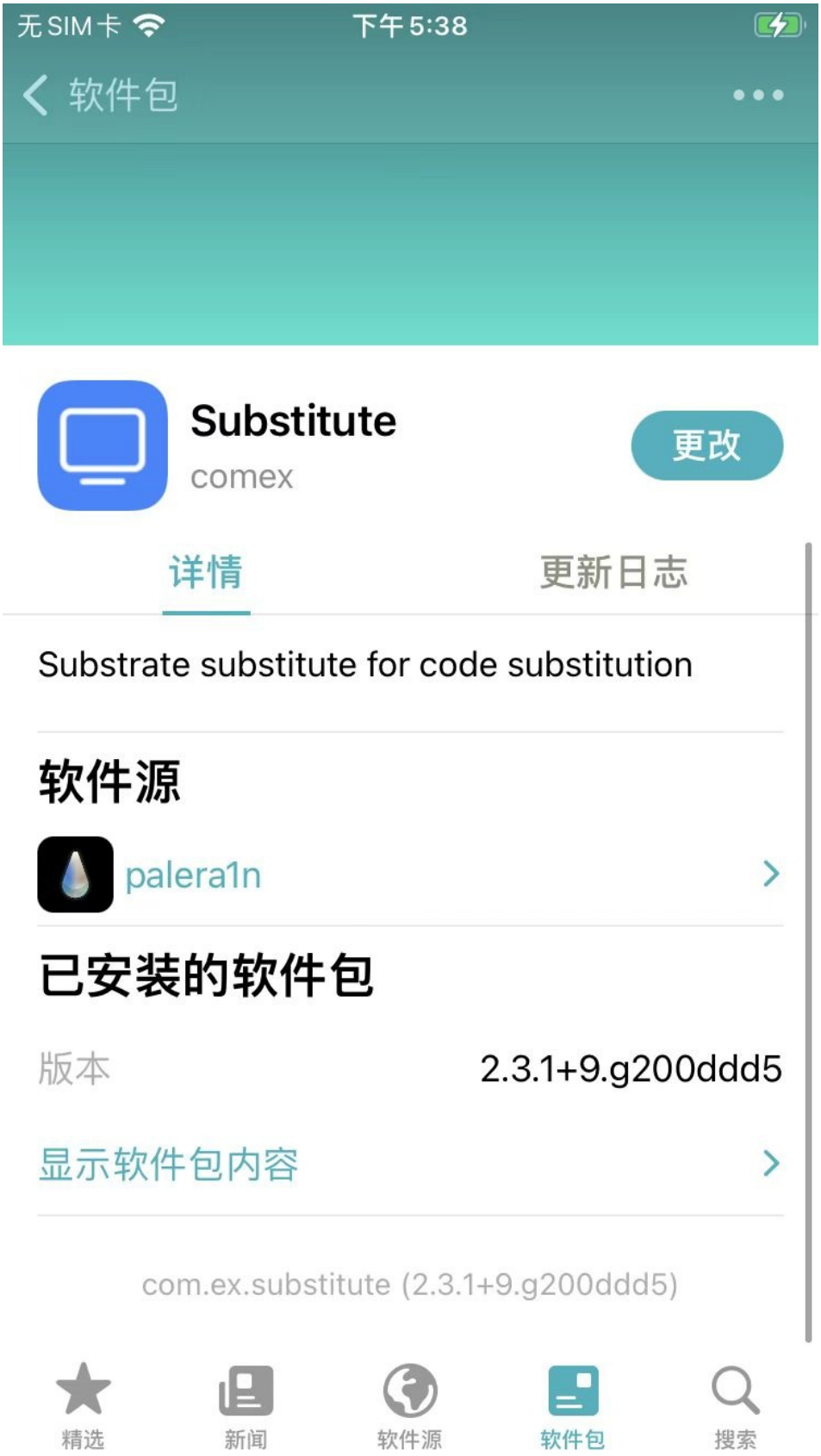

无SIM卡
下午5:37
软件包
Cycript
Jay Freeman (saurik)
更改
详情
更新日志
runtime execution server and disassembler
软件源
palera1n strap
已安装的软件包
版本
0.9.594-procursus1
显示软件包内容
cycript (0.9.594-procursus1)
精选
新闻
软件源
软件包
搜索


无 SIM 卡
下午 5:37
返回
已安装的文件
/
usr/
bin/
cycript
lib/
cycript0.9/
com/
org/
libcycript.cy
libcycript.db
libcycript.dylib
libcycript.0.dylib
精选
新闻
软件源
软件包
搜索

- 说明

- Sileo中安装cycript会自动找到并安装各种依赖
  - adv-cmds
  - 

  - Substitute



- 桌面图标：Substitute

- 

- Substrate Safe Mode

安装Cycript后

可以找到对应二进制文件：

```
iPhone8-150:~ root# which cycript
```

```
/usr/bin/cycript
```

查看基本语法：

```
iPhone8-150:~ root# cycript --help
cycript: unrecognized option `--help'
usage: cycript [-c] [-p <pid|name>] [-r <host:port>] [<script> [<arg ...>]]
```

Cycript的基本用法

```
cycript -p PID_or_AppName
```

进入 `cy#` 开头的命令行界面，即表示注入成功，可以开始调试了

Cycript中常用命令

调试ObjC对象的命令

```
[UIApplication sharedApplication]

UIApp.keyWindow.recursiveDescription().toString()

var topView = [[[[UIApplication sharedApplication] keyWindow] subviews] lastObject]
[topView recursiveDescription].toString()

var p = new Instance(0x157d1e200)
```

打印最顶层页面/窗口

背景知识是，iOS的ObjC的获取最顶层的窗口：

```
[[[[UIApplication sharedApplication] keyWindow] subviews] lastObject]
```

放到Cycript中：

```
[[[[[UIApplication sharedApplication] keyWindow] subviews] lastObject] recursiveDescription].to
String()
```

进一步优化：

写成变量，便于后续引用：

```
var topView = [[[[UIApplication sharedApplication] keyWindow] subviews] lastObject]
[topView recursiveDescription].toString()
```

打印页面详情

已有视图view：

```
cy [[[[UIApplication sharedApplication] keyWindow] subviews] lastObject]
#"<UITransitionView: 0x11f9059e0; frame = (0 0; 375 667); autoresize = W+H; layer = <CALayer: 0
x280129300>>"
```

去打印页面详情，以字符串输出，是：

- 先：`recursiveDescription`
- 再：`toString`

即：

```
var topView = [[[[UIApplication sharedApplication] keyWindow] subviews] lastObject]
[topView recursiveDescription].toString()
```

Cycript使用心得

找按钮的响应函数=处理函数

- 背景

对于页面


无 SIM 卡
下午 4:21
取消
下一步
Apple ID
使用 Apple ID 登录以使用 iCloud 和其他 Apple 服务。
Apple ID
.com
密码
没有或忘记 Apple ID?
Apple ID 是您用于访问所有 Apple 服务的帐户。iCloud 使用无线数据。
您的 Apple ID 信息用于登录时启用 Apple 服务，其中包括 iCloud 云备份，该服务可自动备份设备上的数据，以便需要时进行替换或恢复。您的设备序列号可能被用于检查服务的使用资格。 了解数据的管理方式...

中右上角的 `下一步` 按钮

想要去找，点击之后所触发的对应的处理函数

- 核心思路
- 主要过程和结论

先搞清楚下一步 按钮：

```
<UIButtonLabel: 0x107dacb10; frame = (0 1; 52 20.5); text = '下一步'; opaque = NO; userInteractionEnabled = NO; layer = <_UILabelLayer: 0x2823f0460>>
```

的上2级的元素：

```
<_UIButtonBarButton: 0x107d97600; frame = (0 0; 60 44); tintColor = <UIDynamicSystemColor: 0x2815a6dc0; name = systemBlueColor>; gestureRecognizers = <NSArray: 0x280fefdb0>; layer = <CALayer: 0x2801f8560>>
```

-》对应的：

```
cy# var nextStepBtn2 = #0x107d97600
#"<_UIButtonBarButton: 0x107d97600; frame = (0 0; 60 44); tintColor = <UIDynamicSystemColor: 0x2815a6dc0; name = systemBlueColor>; gestureRecognizers = <NSArray: 0x280fefdb0>; layer = <CALayer: 0x2801f8560>>"

cy# [nextStepBtn2 allTargets]
[NSSet setWithArray:@[#"<_UIButtonBar: 0x283fb7cf0> <_UIButtonBarStackView: 0x107da4c90; frame = (307 6; 60 44); layer = <CALayer: 0x2800665e0>> buttonBar=0x283fb7cf0\nmetrics=0x2815cacc0 layout=0x280f60270 groupLayouts=0x102b1aa70 views=0x280f9af70 guides=0x280f99bc0 activeConstraints=0x280f602d0 minimumInterItemSpace=8.000 minimumInterItemSpaceAnchor=0x2823c34d0 flexibleSpaceEqualSizeAnchor=0x2815c89c0 minimumInterGroupSpaceAnchor=0x2823c1270\nbarButtonGroups={\n<UIBarButtonItemGroup: 0x2823c3070> barButtonItems={\n\t<UIBarButtonItem: 0x107d75ef0> target=0x107d80750 action=_nextButtonSelected: title='\xe4\xb8\x8b\xe4\xb8\x80\xe6\xad\xa5'\n}\n} ",#"<_UIButtonBarTargetAction: 0x2801fbdc0>"]]]
```

中的：

- `_nextButtonSelected:`

Cycript常见问题

偶尔卡死

现象：`cycript -p Preferences` 卡死

原因：偶尔的bug或者其他未知原因

解决办法：多试几次。

包括但不限于：

- 确保设置页面处于前台
- 多运行几次命令
- 打开设置页面，点击进入子页面再返回等等操作

就可以了。

Cycript输出举例

Cycript命令行输出的内容，尤其是对于页面详情，往往输出内容很多。

此处举例说明输出内容大概长什么样：

Cycript输出效果举例

当前页面：


无 SIM 卡
下午 10:55
取消
下一步
Apple ID
使用 Apple ID 登录以使用 iCloud 和其他 Apple 服务。
Apple ID
电子邮件或电话
没有或忘记 Apple ID?
Apple ID 是您用于访问所有 Apple 服务的帐户。iCloud 使用无线数据。
您的 Apple ID 信息用于登录时启用 Apple 服务，其中包括 iCloud 云备份，该服务可自动备份设备上的数据，以便需要时进行替换或恢复。您的设备序列号可能被用于检查服务的使用资格。 了解数据的管理方式...

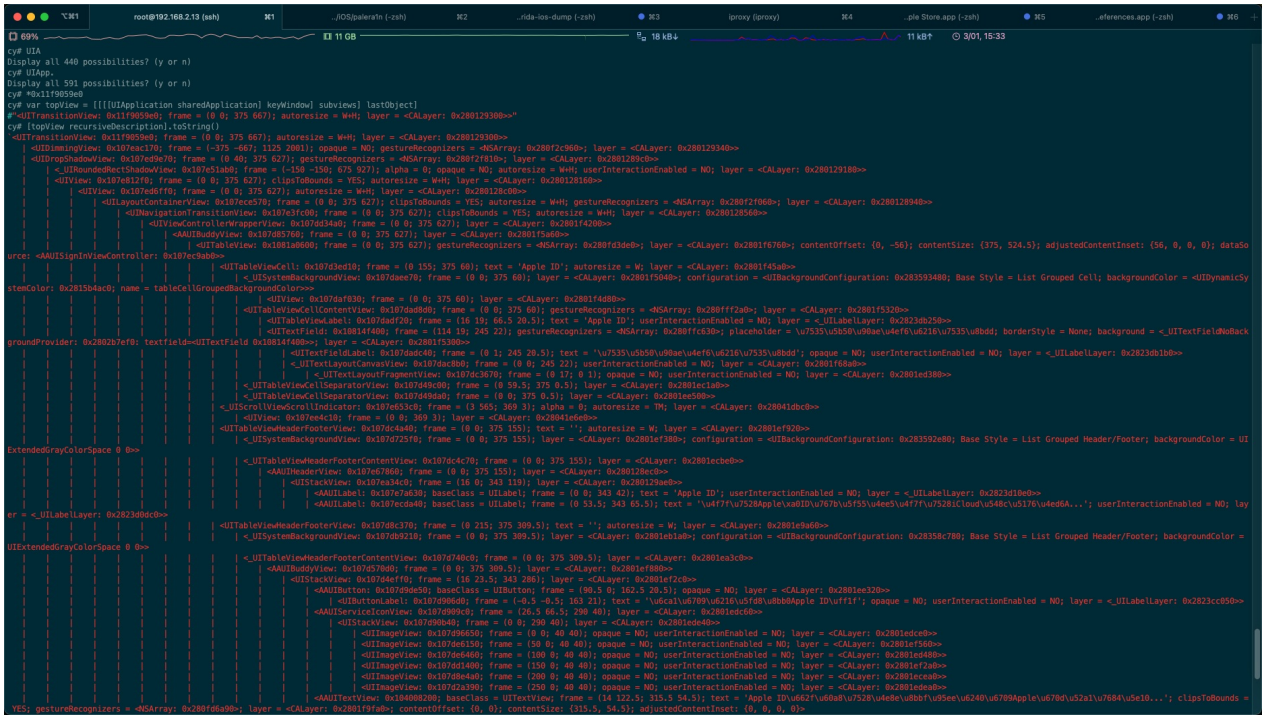
输出结果：

```
cy# var topView = [[[[UIApplication sharedApplication] keyWindow] subviews] lastObject]
#"<UITransitionView: 0x11f9059e0; frame = (0 0; 375 667); autoresize = W+H; layer = <CALayer: 0
x280129300>>"

cy# [topView recursiveDescription].toString()
`<UITransitionView: 0x11f9059e0; frame = (0 0; 375 667); autoresize = W+H; layer = <CALayer: 0x
280129300>>
   | <UIDimmingView: 0x107eac170; frame = (-375 -667; 1125 2001); opaque = NO; gestureRecognize
rs = <NSArray: 0x280f2c960>; layer = <CALayer: 0x280129340>>
   | <UIDropShadowView: 0x107ed9e70; frame = (0 40; 375 627); gestureRecognizers = <NSArray: 0x
280f2f810>; layer = <CALayer: 0x2801289c0>>
   |    | <_UIRoundedRectShadowView: 0x107e51ab0; frame = (-150 -150; 675 927); alpha = 0; opaq
ue = NO; autoresize = W+H; userInteractionEnabled = NO; layer = <CALayer: 0x280129180>>
   |    | <UIView: 0x107e812f0; frame = (0 0; 375 627); clipsToBounds = YES; autoresize = W+H;
layer = <CALayer: 0x280128160>>
   |    |    | <UIView: 0x107ed6ff0; frame = (0 0; 375 627); autoresize = W+H; layer = <CALayer
: 0x280128c00>>
   |    |    |    | <UILayoutContainerView: 0x107ece570; frame = (0 0; 375 627); clipsToBounds =
 YES; autoresize = W+H; gestureRecognizers = <NSArray: 0x280f2f060>; layer = <CALayer: 0x280128
940>>
   |    |    |    |    | <UINavigationTransitionView: 0x107e3fc00; frame = (0 0; 375 627); clip
sToBounds = YES; autoresize = W+H; layer = <CALayer: 0x280128560>>
   |    |    |    |    |    | <UIViewControllerWrapperView: 0x107dd34a0; frame = (0 0; 375 627);
 layer = <CALayer: 0x2801f4200>>
   |    |    |    |    |    |    | <AAUIBuddyView: 0x107d85760; frame = (0 0; 375 627); layer =
<CALayer: 0x2801f5a60>>
   |    |    |    |    |    |    |    | <UITableView: 0x1081a0600; frame = (0 0; 375 627); gest
ureRecognizers = <NSArray: 0x280fd3de0>; layer = <CALayer: 0x2801f6760>; contentOffset: {0, -56}
; contentSize: {375, 524.5}; adjustedContentInset: {56, 0, 0, 0}; dataSource: <AAUISignInViewCo
ntroller: 0x107ec9ab0>>
   |    |    |    |    |    |    |    |    | <UITableViewCell: 0x107d3ed10; frame = (0 155; 375
60); text = 'Apple ID'; autoresize = W; layer = <CALayer: 0x2801f45a0>>
   |    |    |    |    |    |    |    |    |    | <_UISystemBackgroundView: 0x107daee70; frame =
 (0 0; 375 60); layer = <CALayer: 0x2801f5040>; configuration = <UIBackgroundConfiguration: 0x2
83593480; Base Style = List Grouped Cell; backgroundColor = <UIDynamicSystemColor: 0x2815b4ac0;
 name = tableCellGroupedBackgroundColor>>>
   |    |    |    |    |    |    |    |    |    | <UIView: 0x107daf030; frame = (0 0; 375
60); layer = <CALayer: 0x2801f4d80>>
   |    |    |    |    |    |    |    |    |    | <UITableViewCellContentView: 0x107dad8d0; fra
me = (0 0; 375 60); gestureRecognizers = <NSArray: 0x280fff2a0>; layer = <CALayer: 0x2801f5320>>

   |    |    |    |    |    |    |    |    |    |    | <UITableViewLabel: 0x107dadf20; frame = (
16 19; 66.5 20.5); text = 'Apple ID'; userInteractionEnabled = NO; layer = <_UILabelLayer: 0x28
23db250>>
   |    |    |    |    |    |    |    |    |    |    | <UITextField: 0x10814f400; frame = (114
19; 245 22); gestureRecognizers = <NSArray: 0x280ffc630>; placeholder = \u7535\u5b50\u90ae\u4ef6
\u6216\u7535\u8bdd; borderStyle = None; background = <_UITextFieldNoBackgroundProvider: 0x2802b
7ef0: textfield=<UITextField 0x10814f400>>; layer = <CALayer: 0x2801f5300>>
   |    |    |    |    |    |    |    |    |    |    |    | <UITextFieldLabel: 0x107dadc40; fra
me = (0 1; 245 20.5); text = '\u7535\u5b50\u90ae\u4ef6\u6216\u7535\u8bdd'; opaque = NO; userInt
eractionEnabled = NO; layer = <_UILabelLayer: 0x2823db1b0>>
   |    |    |    |    |    |    |    |    |    |    |    | <_UITextLayoutCanvasView: 0x107dac8
b0; frame = (0 0; 245 22); userInteractionEnabled = NO; layer = <CALayer: 0x2801f68a0>>
```

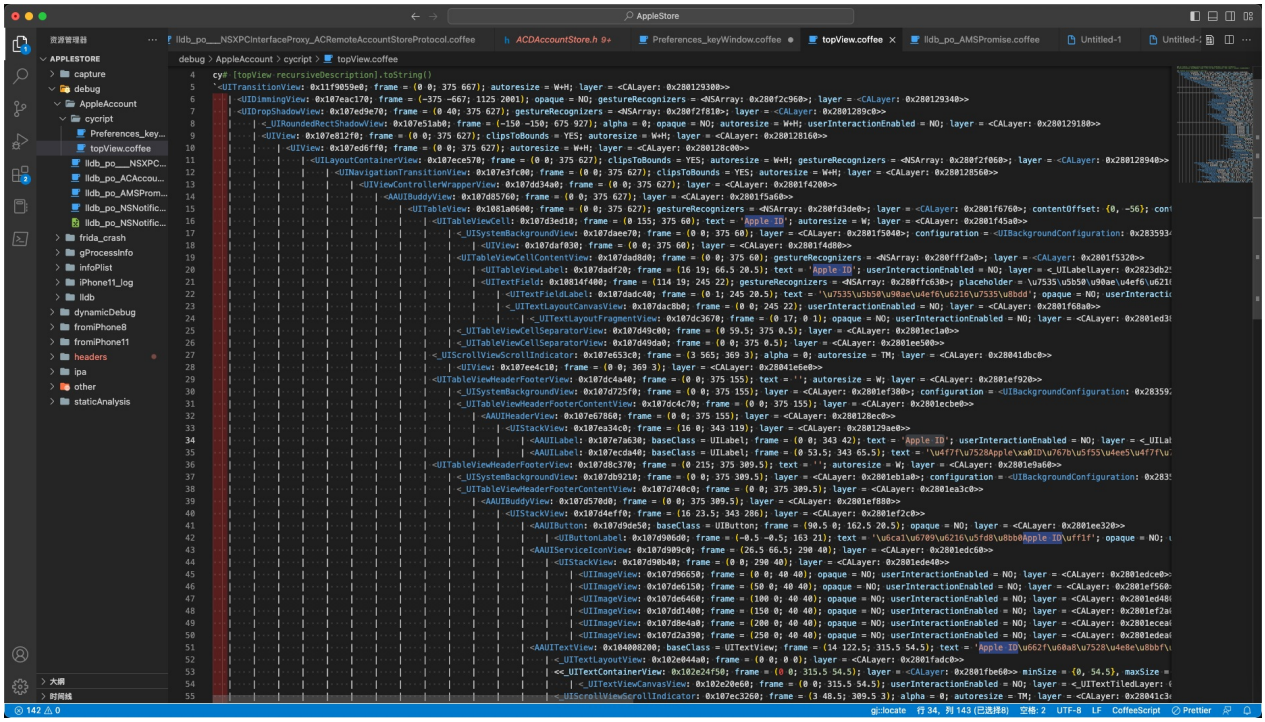
```
   |    |    |    |    |    |    |    |    |    |    |    |    | <_UITextLayoutFragmentView: 0x
107dc3670; frame = (0 17; 0 1); opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x2
801ed380>>
   |    |    |    |    |    |    |    |    |    | <_UITableViewCellSeparatorView: 0x107d49c00;
frame = (0 59.5; 375 0.5); layer = <CALayer: 0x2801ec1a0>>
   |    |    |    |    |    |    |    |    |    | <_UITableViewCellSeparatorView: 0x107d49da0;
frame = (0 0; 375 0.5); layer = <CALayer: 0x2801ee500>>
   |    |    |    |    |    |    |    |    | <_UIScrollViewScrollIndicator: 0x107e653c0; frame =
 (3 565; 369 3); alpha = 0; autoresize = TM; layer = <CALayer: 0x28041dbc0>>
   |    |    |    |    |    |    |    |    |    | <UIView: 0x107ee4c10; frame = (0 0; 369 3); l
ayer = <CALayer: 0x28041e6e0>>
   |    |    |    |    |    |    |    |    | <UITableViewHeaderFooterView: 0x107dc4a40; frame =
(0 0; 375 155); text = ''; autoresize = W; layer = <CALayer: 0x2801ef920>>
   |    |    |    |    |    |    |    |    |    | <_UISystemBackgroundView: 0x107d725f0; frame =
 (0 0; 375 155); layer = <CALayer: 0x2801ef380>; configuration = <UIBackgroundConfiguration: 0x
283592e80; Base Style = List Grouped Header/Footer; backgroundColor = UIExtendedGrayColorSpace 0
 0>>
   |    |    |    |    |    |    |    |    |    | <_UITableViewHeaderFooterContentView: 0x107dc
4c70; frame = (0 0; 375 155); layer = <CALayer: 0x2801ecbe0>>
   |    |    |    |    |    |    |    |    |    |    | <AAUIHeaderView: 0x107e67860; frame = (0
0; 375 155); layer = <CALayer: 0x280128ec0>>
   |    |    |    |    |    |    |    |    |    |    |    | <UIStackView: 0x107ea34c0; frame = (
16 0; 343 119); layer = <CALayer: 0x280129ae0>>
   |    |    |    |    |    |    |    |    |    |    |    |    | <AAUILabel: 0x107e7a630; baseC
lass = UILabel; frame = (0 0; 343 42); text = 'Apple ID'; userInteractionEnabled = NO; layer = <
_UILabelLayer: 0x2823d10e0>>
   |    |    |    |    |    |    |    |    |    |    |    |    | <AAUILabel: 0x107ecda40; baseC
lass = UILabel; frame = (0 53.5; 343 65.5); text = '\u4f7f\u7528Apple\xa0ID\u767b\u5f55\u4ee5\u
4f7f\u7528iCloud\u548c\u5176\u4ed6A...'; userInteractionEnabled = NO; layer = <_UILabelLayer: 0
x2823d0dc0>>
   |    |    |    |    |    |    |    |    | <UITableViewHeaderFooterView: 0x107d8c370; frame =
(0 215; 375 309.5); text = ''; autoresize = W; layer = <CALayer: 0x2801e9a60>>
   |    |    |    |    |    |    |    |    |    | <_UISystemBackgroundView: 0x107db9210; frame =
 (0 0; 375 309.5); layer = <CALayer: 0x2801eb1a0>; configuration = <UIBackgroundConfiguration:
0x28358c780; Base Style = List Grouped Header/Footer; backgroundColor = UIExtendedGrayColorSpac
e 0 0>>
   |    |    |    |    |    |    |    |    |    | <_UITableViewHeaderFooterContentView: 0x107d7
40c0; frame = (0 0; 375 309.5); layer = <CALayer: 0x2801ea3c0>>
   |    |    |    |    |    |    |    |    |    |    | <AAUIBuddyView: 0x107d570d0; frame = (0 0
; 375 309.5); layer = <CALayer: 0x2801ef880>>
   |    |    |    |    |    |    |    |    |    |    |    | <UIStackView: 0x107d4eff0; frame = (
16 23.5; 343 286); layer = <CALayer: 0x2801ef2c0>>
   |    |    |    |    |    |    |    |    |    |    |    |    | <AAUIButton: 0x107d9de50; base
Class = UIButton; frame = (90.5 0; 162.5 20.5); opaque = NO; layer = <CALayer: 0x2801ee320>>
   |    |    |    |    |    |    |    |    |    |    |    |    |    | <UIButtonLabel: 0x107d906
d0; frame = (-0.5 -0.5; 163 21); text = '\u6ca1\u6709\u6216\u5fd8\u8bb0Apple ID\uff1f'; opaque =
 NO; userInteractionEnabled = NO; layer = <_UILabelLayer: 0x2823cc050>>
   |    |    |    |    |    |    |    |    |    |    |    |    | <AAUIServiceIconView: 0x107d90
9c0; frame = (26.5 66.5; 290 40); layer = <CALayer: 0x2801edc60>>
   |    |    |    |    |    |    |    |    |    |    |    |    |    | <UIStackView: 0x107d90b40;
 frame = (0 0; 290 40); layer = <CALayer: 0x2801ede40>>
   |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <UIImageView: 0x107d
96650; frame = (0 0; 40 40); opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x2801
edce0>>
   |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <UIImageView: 0x107d
e6150; frame = (50 0; 40 40); opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x280
```

```
1ef560>>
                                                    | <UIImageView: 0x107d
e6460; frame = (100 0; 40 40); opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x28
01ed480>>
                                                    | <UIImageView: 0x107d
d1400; frame = (150 0; 40 40); opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x28
01ef2a0>>
                                                    | <UIImageView: 0x107d
8e4a0; frame = (200 0; 40 40); opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x28
01ecea0>>
                                                    | <UIImageView: 0x107d
2a390; frame = (250 0; 40 40); opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x28
01edea0>>
                                                    | <AAUITextView: 0x104008200; ba
seClass = UITextView; frame = (14 122.5; 315.5 54.5); text = 'Apple ID\u662f\u60a8\u7528\u4e8e\
u8bbf\u95ee\u6240\u6709Apple\u670d\u52a1\u7684\u5e10...'; clipsToBounds = YES; gestureRecognize
rs = <NSArray: 0x280fd6a90>; layer = <CALayer: 0x2801f9fa0>; contentOffset: {0, 0}; contentSize
: {315.5, 54.5}; adjustedContentInset: {0, 0, 0, 0}>
                                                    | <_UITextLayoutView: 0x102
e044a0; frame = (0 0; 0 0); layer = <CALayer: 0x2801fadc0>>
                                                    | <<_UITextContainerView: 0
x102e24f50; frame = (0 0; 315.5 54.5); layer = <CALayer: 0x2801fbe60>> minSize = {0, 54.5}, max
Size = {315.5, 54.5}, textContainer = <NSTextContainer: 0x283188000 size = (315.500000,38.500000
); widthTracksTextView = YES; heightTracksTextView = YES>; exclusionPaths = 0x1f78db500; lineBr
eakMode = 0>
                                                    | <_UITextViewCanvasVi
ew: 0x102e20e60; frame = (0 0; 315.5 54.5); userInteractionEnabled = NO; layer = <_UITextTiledL
ayer: 0x2832ae7f0>>
                                                    | <_UIScrollViewScrollIndic
ator: 0x107ec3260; frame = (3 48.5; 309.5 3); alpha = 0; autoresize = TM; layer = <CALayer: 0x2
8041c3e0>>
                                                    | <UIView: 0x107eb14b0;
 frame = (0 0; 309.5 3); layer = <CALayer: 0x28041d5e0>>
                                                    | <_UIScrollViewScrollIndic
ator: 0x107e518f0; frame = (309.5 3; 3 48.5); alpha = 0; autoresize = LM; layer = <CALayer: 0x2
8041d600>>
                                                    | <UIView: 0x107ead220;
 frame = (0 0; 3 48.5); layer = <CALayer: 0x28041c2a0>>
                                                    | <OBPrivacyLinkButton: 0x107d6a
250; baseClass = UIButton; frame = (0 193; 343 93); opaque = NO; layer = <CALayer: 0x2801e8a60>>

                                                    | <UIView: 0x107d6a550; fra
me = (0 0; 343 93); userInteractionEnabled = NO; layer = <CALayer: 0x2801ea7c0>>
                                                    | <UITextView: 0x10821
4800; frame = (1 29.5; 341.5 63.5); text = '\u60a8\u7684Apple\xa0ID\u4fe1\u606f\u7528\u4e8e\u76
7b\u5f55\u65f6\u542f\u7528Apple\u670d...'; clipsToBounds = YES; userInteractionEnabled = NO; ge
stureRecognizers = <NSArray: 0x280fe3e40>; layer = <CALayer: 0x2801eaf60>; contentOffset: {0, 0}
; contentSize: {341.5, 63.5}; adjustedContentInset: {0, 0, 0, 0}>
                                                    | <_UITextLayoutV
iew: 0x107d77150; frame = (0 0; 0 0); layer = <CALayer: 0x2801e8120>>
                                                    | <<_UITextContai
nerView: 0x107d76da0; frame = (0 0; 341.5 63.5); layer = <CALayer: 0x2801e8160>> minSize = {0,
63.5}, maxSize = {341.5, 63.5}, textContainer = <NSTextContainer: 0x2831b3160 size = (341.500000
,63.500000); widthTracksTextView = YES; heightTracksTextView = YES>; exclusionPaths = 0x1f78db5
00; lineBreakMode = 0>
                                                    | <_UITextVi
```

```
ewCanvasView: 0x107d76fb0; frame = (0 0; 341.5 63.5); userInteractionEnabled = NO; layer = <_UI
TextTiledLayer: 0x28328b840>>
                                                                                    <_UIScrollViewS
crollIndicator: 0x11f9154a0; frame = (3 57.5; 335.5 3); alpha = 0; autoresize = TM; layer = <CA
Layer: 0x28041d740>>
                                                                                        <UIView: 0
x107eafe80; frame = (0 0; 335.5 3); layer = <CALayer: 0x28041c080>>
                                                                                    <_UIScrollViewS
crollIndicator: 0x107ed65a0; frame = (335.5 3; 3 57.5); alpha = 0; autoresize = LM; layer = <CA
Layer: 0x28041c420>>
                                                                                        <UIView: 0
x107e9c4a0; frame = (0 0; 3 57.5); layer = <CALayer: 0x28041d900>>
                                                                                <OBTintInheritingIma
geView: 0x107d77b30; baseClass = UIImageView; frame = (156.5 0; 30 24); opaque = NO; userIntera
ctionEnabled = NO; tintColor = <UIDynamicSystemColor: 0x2815a6dc0; name = systemBlueColor>; lay
er = <CALayer: 0x2801eb9e0>>
                                                                            <_UIScrollViewScrollIndicator: 0x107ec93b0; frame =
 (369 105.5; 3 462.5); alpha = 0; autoresize = LM; layer = <CALayer: 0x28041f0c0>>
                                                                                <UIView: 0x107e1b0d0; frame = (0 0; 3 462.5);
 layer = <CALayer: 0x28041e5e0>>
                                                                        <UINavigationBar: 0x107e957e0; frame = (0 0; 375 56); opaque = NO; aut
oresize = W; layer = <CALayer: 0x2801288e0>>
                                                                            <_UIBarBackground: 0x107eadd90; frame = (0 0; 375 56); userIntera
ctionEnabled = NO; layer = <CALayer: 0x2801288c0>>
                                                                                <UIImageView: 0x107d99450; frame = (0 0; 375 56); alpha = 0;
 opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x2801f4360>>
                                                                            <_UIBarBackgroundShadowView: 0x11f908b20; frame = (0 56; 375
0); layer = <CALayer: 0x280129440>> clientRequestedContentView effect=none
                                                                                <_UIBarBackgroundShadowContentImageView: 0x107eaa210; f
rame = (0 0; 375 0); alpha = 0; opaque = NO; autoresize = W+H; userInteractionEnabled = NO; lay
er = <CALayer: 0x280129100>>
                                                                            <_UINavigationBarContentView: 0x107ed4fa0; frame = (0 0; 375 56);
 layer = <CALayer: 0x280128880>> layout=0x107e1a430
                                                                                <_UIButtonBarStackView: 0x107df1d90; frame = (8 6; 46 44); l
ayer = <CALayer: 0x2801f7520>> buttonBar=0x283fbc5a0
                                                                                <_UIButtonBarButton: 0x107de7110; frame = (0 0; 46 44);
 tintColor = <UIDynamicSystemColor: 0x2815a6dc0; name = systemBlueColor>; gestureRecognizers = <
NSArray: 0x280f2a3a0>; layer = <CALayer: 0x2801f4280>>
                                                                                <_UIModernBarButton: 0x107ef1fd0; frame = (8 11; 38
 21.5); opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x280129b20>>
                                                                                <UIButtonLabel: 0x107ed0070; frame = (0 1; 35
20.5); text = '\u53d6\u6d88'; opaque = NO; userInteractionEnabled = NO; layer = <_UILabelLayer:
 0x2823d26c0>>
                                                                            <_UIButtonBarStackView: 0x107dccc80; frame = (307 6; 60 44);
 layer = <CALayer: 0x2801f4440>> buttonBar=0x283fbc4b0
                                                                                <_UIButtonBarButton: 0x107d2b8b0; frame = (0 0; 60 44);
 tintColor = <UIDynamicSystemColor: 0x2815a6dc0; name = systemBlueColor>; gestureRecognizers = <
NSArray: 0x280fd3420>; layer = <CALayer: 0x2801f7280>>
                                                                                <_UIModernBarButton: 0x107db1ef0; frame = (0 11; 52
 21.5); opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x2801f5f00>>
                                                                                <UIButtonLabel: 0x107d37c60; frame = (0 1; 52
20.5); text = '\u4e0b\u4e00\u6b65'; opaque = NO; userInteractionEnabled = NO; layer = <_UILabel
Layer: 0x2823d9f40>>
                                                                            <UIView: 0x107e097a0; frame = (0 0; 0 0); userInteractionEnabled =
 NO; layer = <CALayer: 0x2801296e0>>`
cy#
```



拷贝到VSCode中的显示效果：



LLDBTools

TODO：

- 【已解决】用MonkeyDev的LLDBTools去打印UI界面元素

---

iOS逆向调试界面元素时，也可以用：MonkeyDev的 `LLDBTools` 的相关命令，输出界面元素信息。

chisel

`chisel` 本身是 Xcode内置调试器：LLDB 的插件，其中有部分命令，也可以用来，调试打印界面元素。

FLEX

TODO：

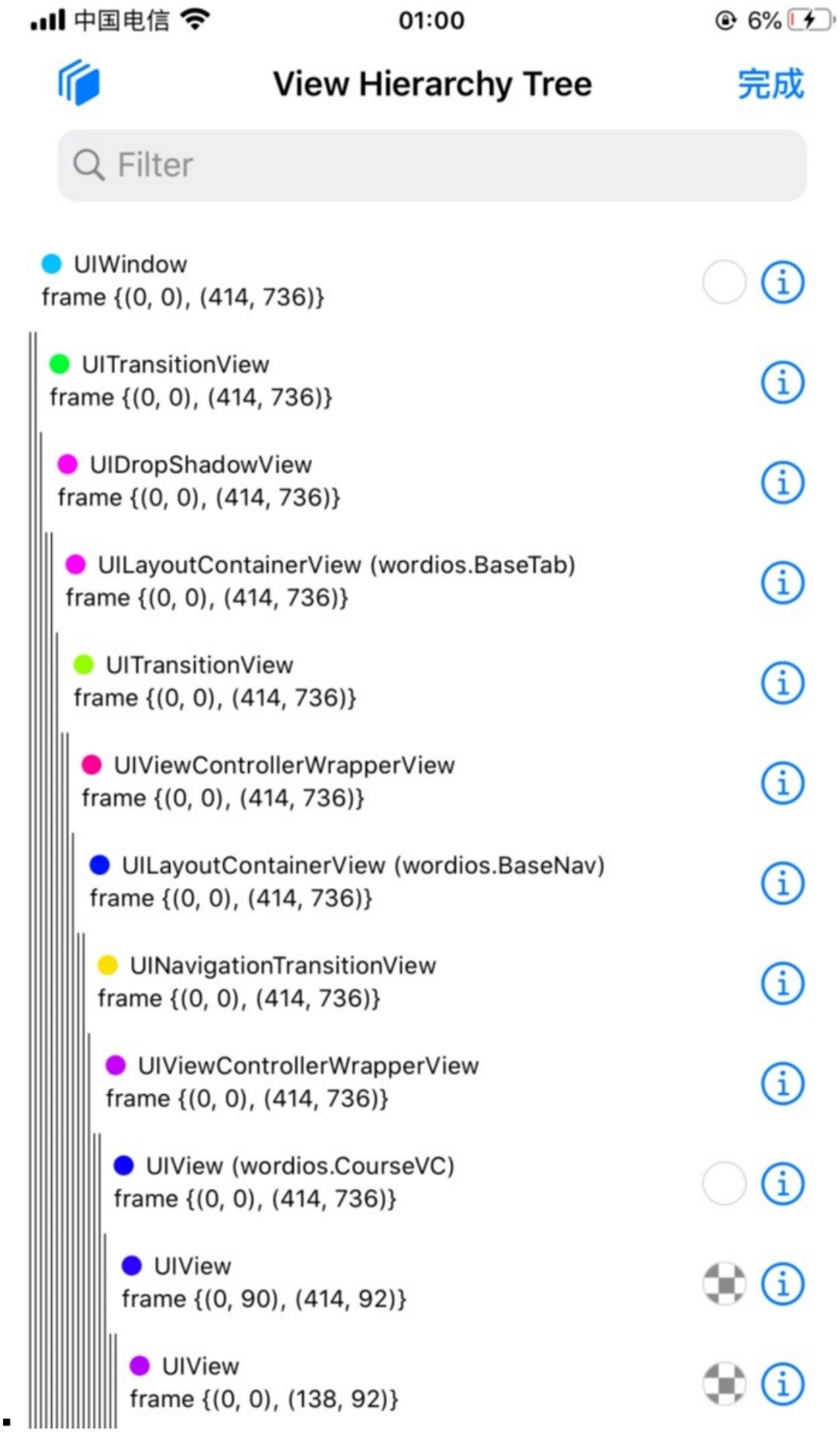
【整理】iOS的iPhone越狱和改机相关知识

---

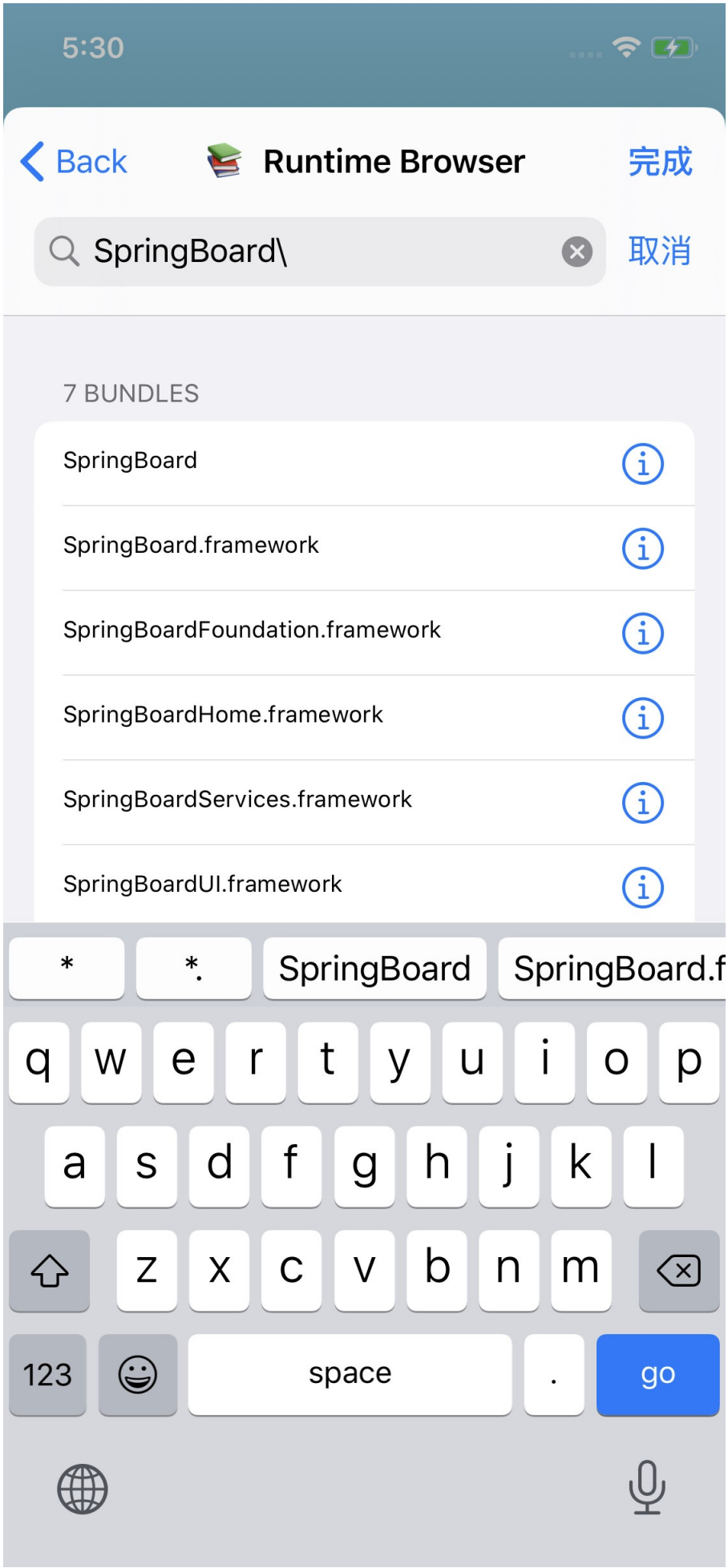
iOS越狱插件 `FLEX` ，可以用来辅助调试iOS的app的界面元素。

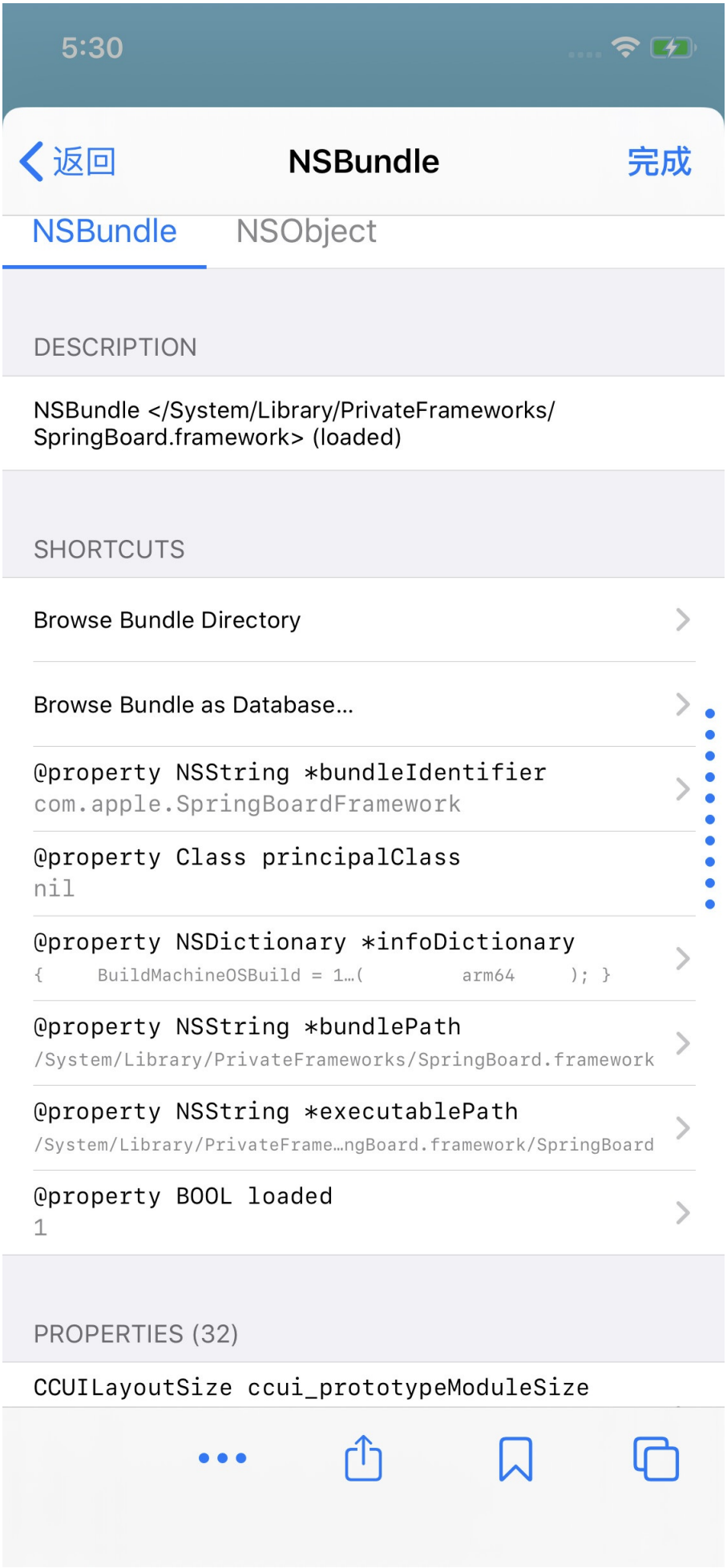
- `FLEX`
  - 效果
    - 当它加载时，会向目标程序上方添加一个悬浮的工具栏，通过这个工具栏可以查看和修改视图的层级结构、动态修改类的属性、动态调用实例和方法、动态查看类和框架以及动态修改UI等。
  - 截图
    -


中国电信
01:00
6%
教程
menu
views
select
recent
close
Word 文档
Excel 表格
PPT 幻灯片
热门教程
2009年同类图书销量统计
将文档中的表格转化为...
存档方便快捷
如何让你的图表动起来
让你的工作报表更新颖
个人简介
姓名
性别
照片
出生年月
籍贯
家庭住址
婚姻状况
01 语法:
简单易懂不会忘，1个逻辑套用10个公式
用表格也能制作简历
快速晋升商务办公达人
震惊！PPT究竟有多么...
PPT 进阶
首页
文档
模版
教程
我的
@稀土掘金技术社区

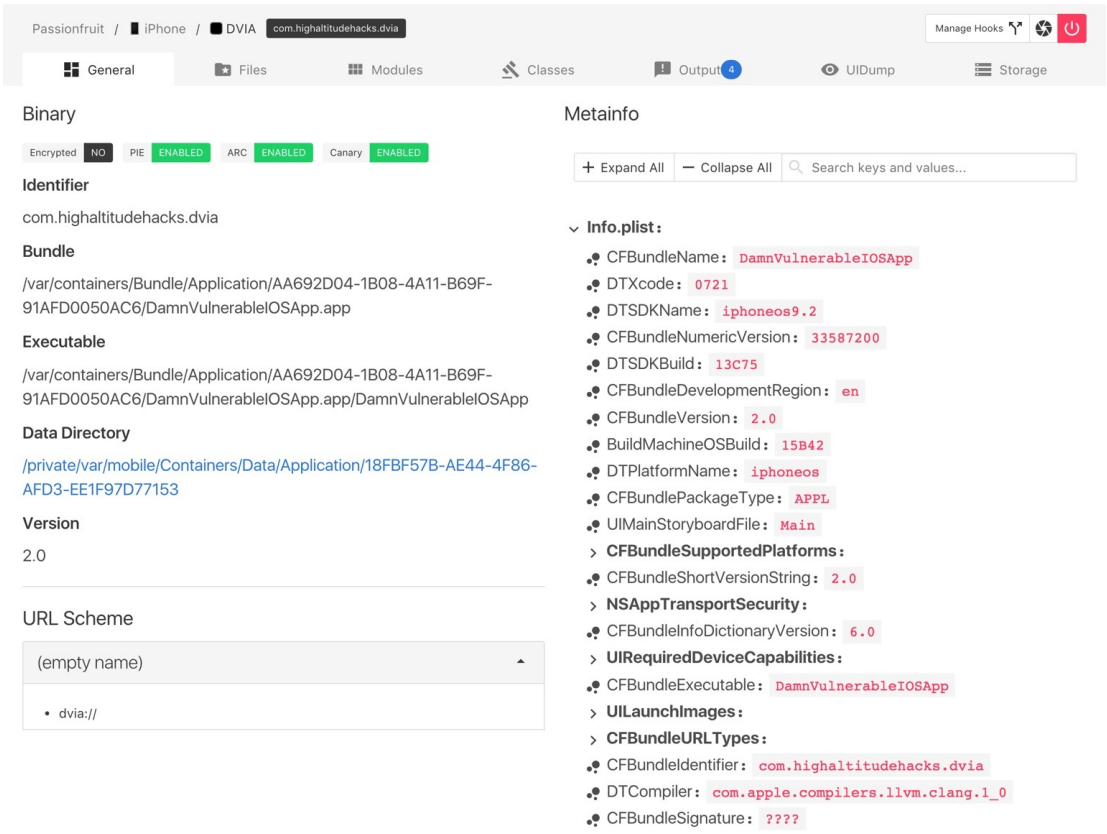

中国电信
01:00
6%
View Hierarchy Tree
完成
Filter
UIWindow
frame {(0, 0), (414, 736)}
UITransitionView
frame {(0, 0), (414, 736)}
UIDropShadowView
frame {(0, 0), (414, 736)}
UILayoutContainerView (wordios.BaseTab)
frame {(0, 0), (414, 736)}
UITransitionView
frame {(0, 0), (414, 736)}
UIViewControllerWrapperView
frame {(0, 0), (414, 736)}
UILayoutContainerView (wordios.BaseNav)
frame {(0, 0), (414, 736)}
UINavigationTransitionView
frame {(0, 0), (414, 736)}
UIViewControllerWrapperView
frame {(0, 0), (414, 736)}
UIView (wordios.CourseVC)
frame {(0, 0), (414, 736)}
UIView
frame {(0, 90), (414, 92)}
UIView
frame {(0, 0), (138, 92)}

- 好像还可以擦好看类的定义
  -


5:30
Back
Runtime Browser
完成
SpringBoard\
取消
7 BUNDLES
SpringBoard
SpringBoard.framework
SpringBoardFoundation.framework
SpringBoardHome.framework
SpringBoardServices.framework
SpringBoardUI.framework
*
*.
SpringBoard
SpringBoard.f
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
123
space
.
go


5:30
返回
NSBundle
完成
NSBundle
NSObject
DESCRIPTION
NSBundle </System/Library/PrivateFrameworks/SpringBoard.framework> (loaded)
SHORTCUTS
Browse Bundle Directory
Browse Bundle as Database…
@property NSString *bundleIdentifier
com.apple.SpringBoardFramework
@property Class principalClass
nil
@property NSDictionary *infoDictionary
{ BuildMachineOSBuild = 1…(arm64); }
@property NSString *bundlePath
/System/Library/PrivateFrameworks/SpringBoard.framework
@property NSString *executablePath
/System/Library/PrivateFrame…ngBoard.framework/SpringBoard
@property BOOL loaded
1
PROPERTIES (32)
CCUILayoutSize ccui_prototypeModuleSize

Passionfruit

- `Passionfruit`
  - 概述：
    - `Passionfruit` 通过 `frida` 注入代码到目标应用实现功能，再通过 `node.js` 服务端消息代理与浏览器通信，用户通过访问网页即可对App实现常规的检测任务
    - `Passionfruit` 最大特点就是基于Web的图形界面，所以服务端支持跨平台的。
      - 在不少界面都添加了搜索功能，如模块列表、导出符号、 `Objective-C` 类，甚至 `Plist` 这样的序列化数据
  - 截图
    - 

  - Github
    - chaitin/passionfruit: [WIP] Crappy iOS app analyzer
      - Simple iOS app blackbox assessment tool. Powered by frida.re and vuejs.
    - 注：2021年停止维护了
    - wiki
      - Screenshots · chaitin/passionfruit Wiki (github.com)

动态调试心得

TODO：

- 计算类的属性的偏移量
  - 【已解决】调试寻找HAMPlayerInternal的_currentTime中字段的偏移量
  - 【整理】iOS逆向心得：通过查看类的地址保存的值找到值和属性字段的偏移量和对应关系
  - 【已解决】iOS逆向：写hook代码时打印出类的私有属性变量值的类型
  - 【整理】iOS逆向心得：类的属性字段偏移量计算要加上isa的父类
- C++
  - 【整理】iOS逆向心得：Cronet相关的C++的struct结构体类的属性和函数的偏移量计算逻辑
  - 【已解决】iOS逆向：IDA中如何逆向分析C++的vtable
  - 【整理】iOS逆向涉及内容：C++中的vtable
- 其他
  - 【已解决】iOS逆向：查看NSMutableURLRequest的HTTPBody的data
  - 【已解决】Xcode中调试iOS的app再次报错：Thread EXC_BREAKPOINT code 1 subcode 0x1bf09c598

---

iOS逆向的动态调试，有很多心得，整理如下。

Xcode相关

TODO：

- Xcode相关
  - 问题
    - 【已解决】XCode调试YouTube报错：Unable to install There was an internal API error
  - 心得
    - 【已解决】XCode+MonkeyDev调试YouTube：如何在广告页面停止供调试

---

lldb

- 【整理】iOS逆向心得：lldb中打印d的d0寄存器不是double而是data
- 【整理】如何找到Xcode中lldb调试出的无名的函数对应的IDA的伪代码中是哪个函数

EXC_BREAKPOINT

- EXC_BREAKPOINT
  - 【已解决】iOS逆向调试报错EXC_BREAKPOINT：HAMNetworkRequestResponseEvent的initWithRequest
  - 【已解决】Xcode中调试iOS的app再次报错：Thread EXC_BREAKPOINT code 1 subcode 0x1bf09c598
  - 【未解决】Xcode调试iOS的YouTube时objc_msgSend崩溃：Thread EXC_BREAKPOINT code 1 subcode 0x1bf09c598

ObjC

TODO：

- ObjC
  - 【整理】iOS逆向调试心得：ObjC或ARM中从偏移量中取值的不同写法
  - 【未解决】Xcode调试iOS的Objc时获取self的父类的实例
  - 【整理】iOS逆向心得：ObjC函数调用时参数顺序和汇编代码中寄存器传递的参数顺序不一致
  - 【整理】iOS逆向和IDA使用心得：调用objc_msgSend时传递给MLPlayerItemQOEErrorEvent的initWithError:fatal:absoluteTime:的参数不够
  - 【整理】iOS逆向心得：打印ObjC类的属性

objc_msgSend

iOS的ObjC的函数调用，比如A函数调用B函数，底层都是通过 `objc_msgSend` 实现的。

所以iOS逆向期间，涉及到最多的，应该就属 `objc_msgSend` 了。

所以关于 `objc_msgSend` 也有很多心得，整理如下。

不带lldb_unnamed_symbol的无名的bl，往往是更重要的，我们所关注的objc_msgSend

折腾：

【未解决】研究抖音关注逻辑：___lldb_unnamed_symbol1588524$$AwemeCore

期间，调试到目前的心得：

如果是带 `___lldb_unnamed_symbol` 的写法，往往不是主要的，我们所关心的 `objc_msgSend` 函数

而无名的 `bl` ，往往是重要的，我们所关注的： `objc_msgSend` 的相关调用

举例：

```
    0x11427c2c8 <+336>: bl     0x115ce58fc
```

其实就是： `objc_msgSend`

而其他很多其他的bl：

```
    0x11427c2b0 <+312>: bl     0x11427e920               ; ___lldb_unnamed_symbol1588573$$AwemeC
ore
```

只是个 `jmp_objc_retain` ，不是我们关注的重点。

Runtime运行时

TODO：

- ObjC运行时
  - 【记录】iOS中的ObjC的函数：dispatch_async
  - 【已解决】iOS逆向心得：OS_dispatch_data
  - 【已解决】iOS底层函数：objc_enumerationMutation

po

TODO:

- po失效
  - 【无需解决】Xcode中lldb调试iOS的ObjC的汇编代码时：偶尔po失效打印不出变量类型
  - 【未解决】Xcode中lldb的po再次失效尝试搞懂内部原因

LLDB中的调试命令po，也是iOS逆向期间，用的最多的命令：用于查看某个地址，具体是什么（iOS的ObjC的）类。

对于 `po`，也有很多经验和心得，整理如下。

类Class 对比 实例Instance

折腾：

【记录】XCode+MonkeyDev动态调试YouTube类：YTWatchMiniBarViewController

期间，可以通过hook代码：

```
%hook YTWatchController

//- (void)playbackControllerDidLoadPlayerWithPlaybackData:(id)arg1;
- (void)playbackControllerDidLoadPlayerWithPlaybackData:(id)arg1{
  iosLogInfo("arg1=%@", arg1);
  %orig;
}

%end
```

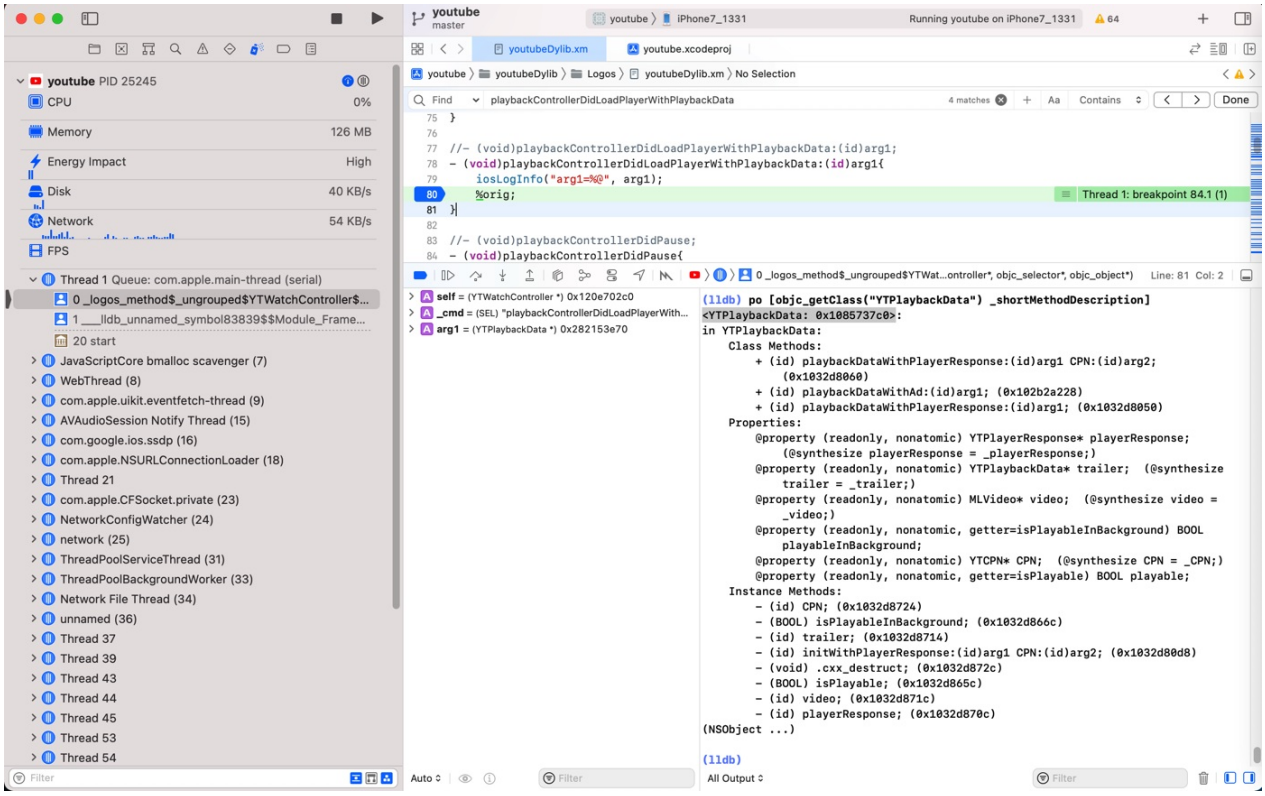
而输出log：

```
2022-03-27 17:16:50.049397+0800 YouTube[25245:2617390] hook_ youtubeDylib.xm YTWatchController$
playbackControllerDidLoadPlayerWithPlaybackData$ arg1 <YTPlaybackData: 0x282153e70>
```

而另外可以通过 `po` 查看到类 `YTPlaybackData` 的信息

```
(lldb) po [objc_getClass("YTPlaybackData") _shortMethodDescription]
<YTPlaybackData: 0x1085737c0>:
in YTPlaybackData:
   Class Methods:
      + (id) playbackDataWithPlayerResponse:(id)arg1 CPN:(id)arg2; (0x1032d8060)
      + (id) playbackDataWithAd:(id)arg1; (0x102b2a228)
      + (id) playbackDataWithPlayerResponse:(id)arg1; (0x1032d8050)
   Properties:
     @property (readonly, nonatomic) YTPlayerResponse* playerResponse; (@synthesize playerRespo
nse = _playerResponse;)
```

```
    @property (readonly, nonatomic) YTPlaybackData* trailer; (@synthesize trailer = _trailer;)
    @property (readonly, nonatomic) MLVideo* video; (@synthesize video = _video;)
    @property (readonly, nonatomic, getter isPlayableInBackground) BOOL playableInBackground;
    @property (readonly, nonatomic) YTCPN* CPN; (@synthesize CPN = _CPN;)
    @property (readonly, nonatomic, getter isPlayable) BOOL playable;
Instance Methods:
    - (id) CPN; (0x1032d8724)
    - (BOOL) isPlayableInBackground; (0x1032d866c)
    - (id) trailer; (0x1032d8714)
    - (id) initWithPlayerResponse:(id)arg1 CPN:(id)arg2; (0x1032d80d8)
    - (void) .cxx_destruct; (0x1032d872c)
    - (BOOL) isPlayable; (0x1032d865c)
    - (id) video; (0x1032d871c)
    - (id) playerResponse; (0x1032d870c)
(NSObject ...)
```



两者对比：

- `Class` = 类
  - 无需遇到对应类的变量，任何时候，只要代码加载到内存了，即可查看具体的内容
  - 查看类的信息的方式：

    ```
    po [objc_getClass("YTPlaybackData") _shortMethodDescription]
    ```

  - 场景举例
    - 比如给 `YouTube` 加了断点 `UIApplicationMain`，断点生效时，即可查看类 `YTPlaybackData` 的信息，而无需实际调试找到 `YTPlaybackData` 的实例变量
- `Instance` = 实例
  - 只有遇到对应的变量类型了，才能看到具体的值

        - 比如，此处是运行到函数playbackControllerDidLoadPlayerWithPlaybackData的内部，YTPlaybackData作为参数，所以才能看到具体的Instance实例的值
    - 查看示例变量值的方式
        - 举例

```
(lldb) po [(YTPlaybackData *)0x282153e70 isPlayable]
true

(lldb) po [(YTPlaybackData *)0x282153e70 isPlayableInBackground]
false

(lldb) po [(YTPlaybackData *)0x282153e70 video]
 MLVideo: 0x282149da0>

(lldb) po [(YTPlaybackData *)0x282153e70 trailer]
nil
```

            - 即可看到，当前类 `YTPlaybackData` 的实例：`< 0x282153e70>` 的各种属性值

用po打印Class类的属性Property或函数Method

- 打印Class的属性或函数值

```
po [ClassName classMethodOrProperty]
```

- 打印Class的Instance的属性或函数之

```
po [objc_getClass("ClassName") instanceMethodOrProperty]
```

举例：

```
(lldb) po [objc_getClass("TTMacroManager") _shortMethodDescription]
<TTMacroManager: 0x103b56e48>:
in TTMacroManager:
    Class Methods:
        + (BOOL) isBDWEBIMAGE_APP_EXTENSIONS; (0x117cf1d50)
        + (BOOL) isDebug; (0x117cf1e28)
(NSObject ...)
```

进一步的，对应着（导出抖音的）头文件：

```
#import <objc/NSObject.h>

@interface TTMacroManager : NSObject
{
}

+ (_Bool)isBDWEBIMAGE_APP_EXTENSIONS;
+ (_Bool)isDebug;
```

```
@end
```

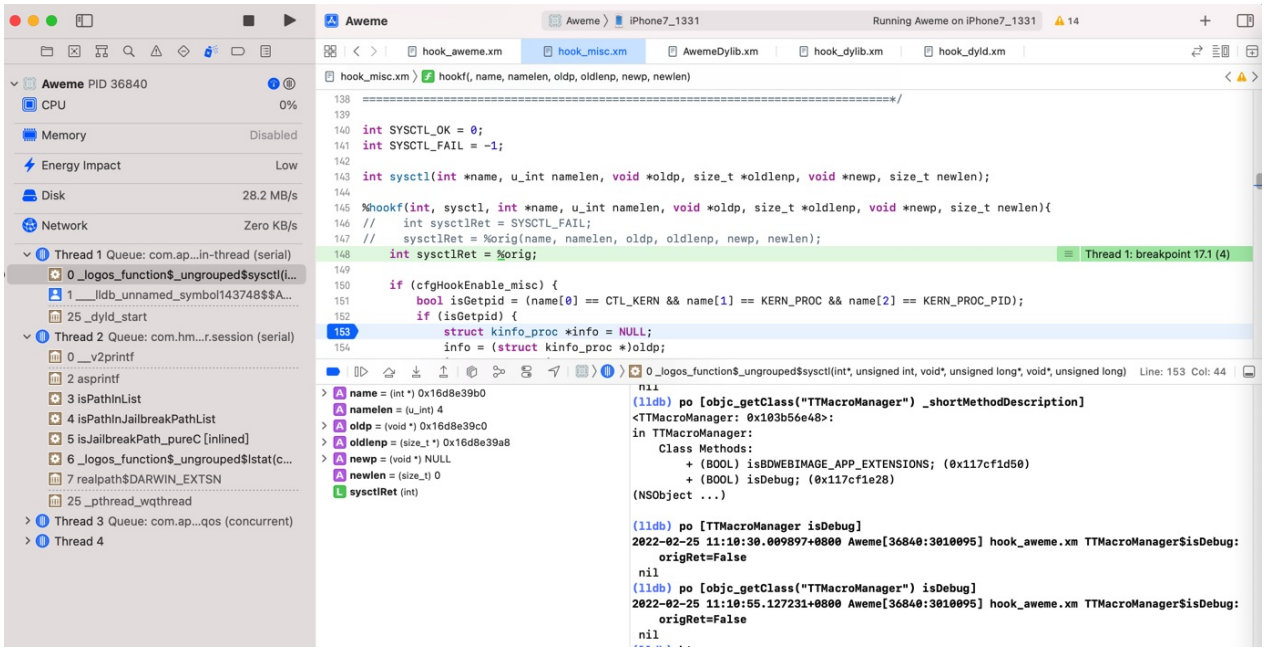
想要查看当前的Class的值，则是：

```
(lldb) po [TTMacroManager isDebug]
2022-02-25 11:10:30.009897+0800 Aweme[36840:3010095] hook_aweme.xm TTMacroManager$isDebug  origRet=False
 nil
```

注：此处输出的是被我加了hook了的代码的log

如果想要查看实例instance的值，则是：

```
(lldb) po [objc_getClass("TTMacroManager") isDebug]
2022-02-25 11:10:55.127231+0800 Aweme[36840:3010095] hook_aweme.xm TTMacroManager$isDebug  origRet=False
 nil
```



po 失效时换用 object_getClassName 查看是什么类

iOS逆向期间，正常的话， po 是可以打印出某个地址，具体是什么（ObjC的）类

比如：

```
(lldb) po 0x0000000137419800
<AWEInviteSearchTableViewCell: 0x137419800; baseClass = UITableViewCell; frame = (0 152; 375 76)
; autoresize = W; layer = <CALayer: 0x28f694b00>>
```

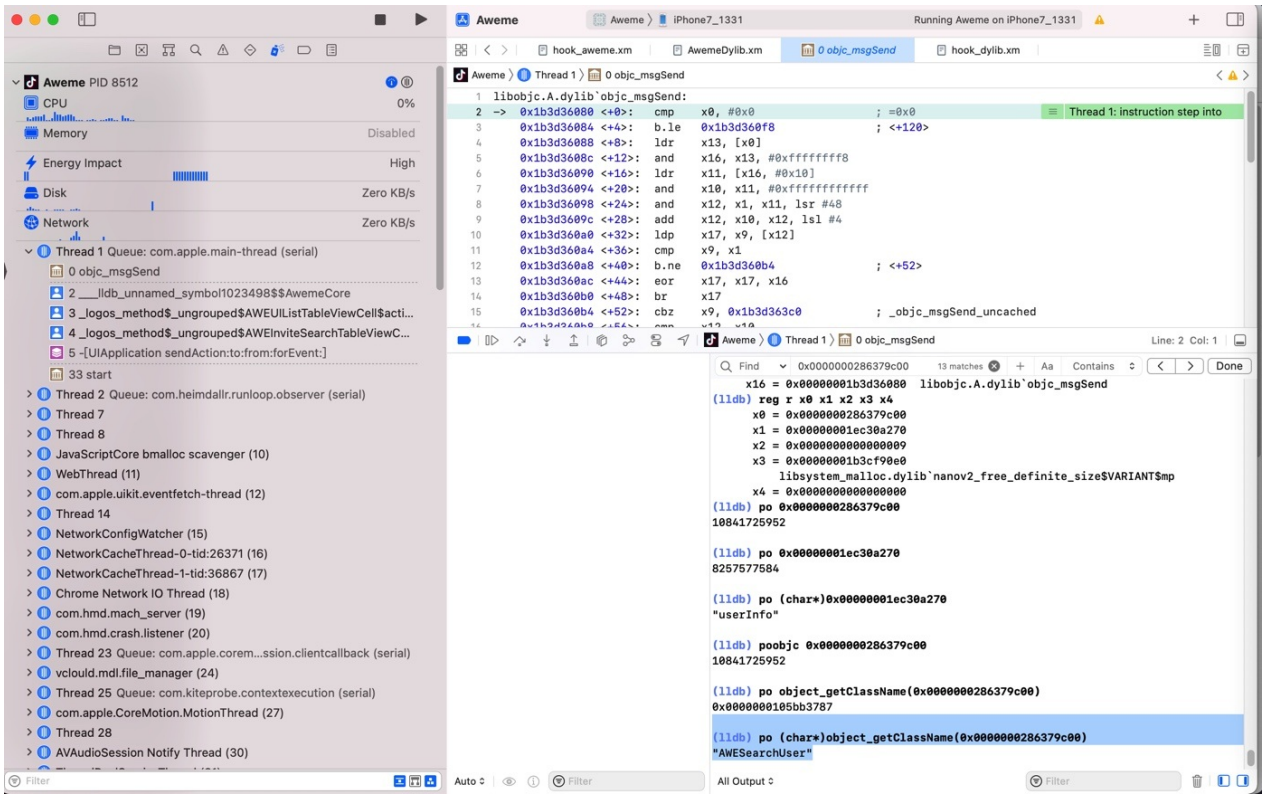
-> 从而通过调试搞懂代码的具体逻辑：调用了什么(ObjC的)类的什么函数。

而有时候，不知何故， `po` 失效，则打印不出来是什么类：

```
(lldb) po 0x0000000286379c00
10841725952
```

此时，可以换用： `object_getClassName`

```
(lldb) po (char *)object_getClassName(0x0000000286379c00)
"AWESearchUser"
```



po查看类的描述的同时可以看到父类的相关定义

比如抖音的：

`Aweme_classDump/Aweme_17.8.0_header/Aweme/AWEPlayInteractionFollowSuccessElement.h`

```
#import "AWEPlayInteractionBottomElement.h"

#import "AWEUserMessage-Protocol.h"

@class AWEAntiAddictedNoticeBarView, AWEHistoryPublicDataController, NSString;

@interface AWEPlayInteractionFollowSuccessElement : AWEPlayInteractionBottomElement <AWEUserMes
sage>
{
    AWEAntiAddictedNoticeBarView *_antiAddictedNoticeBarView;
    AWEHistoryPublicDataController *_listDataController;
    long long _actionType;
}
```

```
- (void).cxx_destruct;
@property(nonatomic) long long actionType; // @synthesize actionType=_actionType;
@property(retain, nonatomic) AWEHistoryPublicDataController *listDataController; // @synthesize
 listDataController=_listDataController;
@property(retain, nonatomic) AWEAntiAddictedNoticeBarView *antiAddictedNoticeBarView; // @synth
esize antiAddictedNoticeBarView=_antiAddictedNoticeBarView;
- (void)didFinishUnFollowUser:(id)arg1 status:(long long)arg2 error:(id)arg3;
- (void)p_hideAntiAddictedNoticeBarView:(long long)arg1 animation:(CDUnknownBlockType)arg2;
- (void)p_showAntiAddictedNoticeBarViewWithCompletion:(CDUnknownBlockType)arg1;
- (void)noticeTapped;
- (void)showFollowSuccessNoticeBar:(id)arg1;
- (void)hideMutexTempElement:(CDUnknownBlockType)arg1;
- (void)dealloc;
- (void)viewDidDisposed;
- (void)reset;
- (void)viewDidLoad;
- (void)initializeElement;

// Remaining properties
@property(readonly, copy) NSString *debugDescription;
@property(readonly, copy) NSString *description;
@property(readonly) unsigned long long hash;
@property(readonly) Class superclass;

@end
```

去Xcode的lldb中动态调试抖音期间，通过：

```
po [objc_getClass("AWEPlayInteractionFollowSuccessElement") _shortMethodDescription]
```

- 不仅能看到：类 `AWEPlayInteractionFollowSuccessElement` 本身的信息
- 还能看到：父类 `AWEPlayInteractionBottomElement`
  - 父类的父类：`AWEPlayInteractionBottomElement`
    - 父类的父类的父类：`AWEPlayInteractionBaseElement`
      - 父类的父类的父类的父类：`AWEBaseElement`
        - 直到最后的根对象：`NSObject`

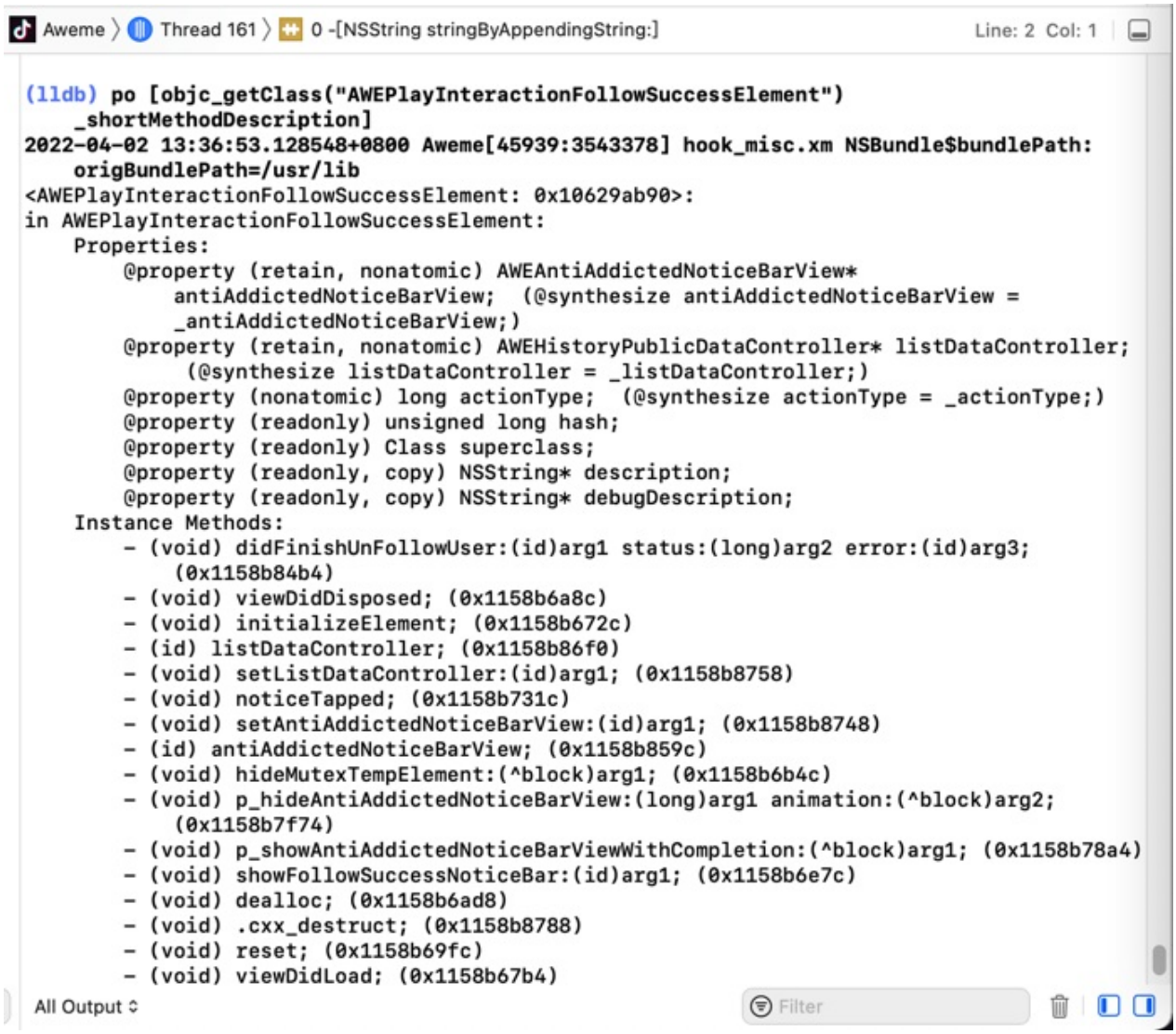
具体输出内容是：

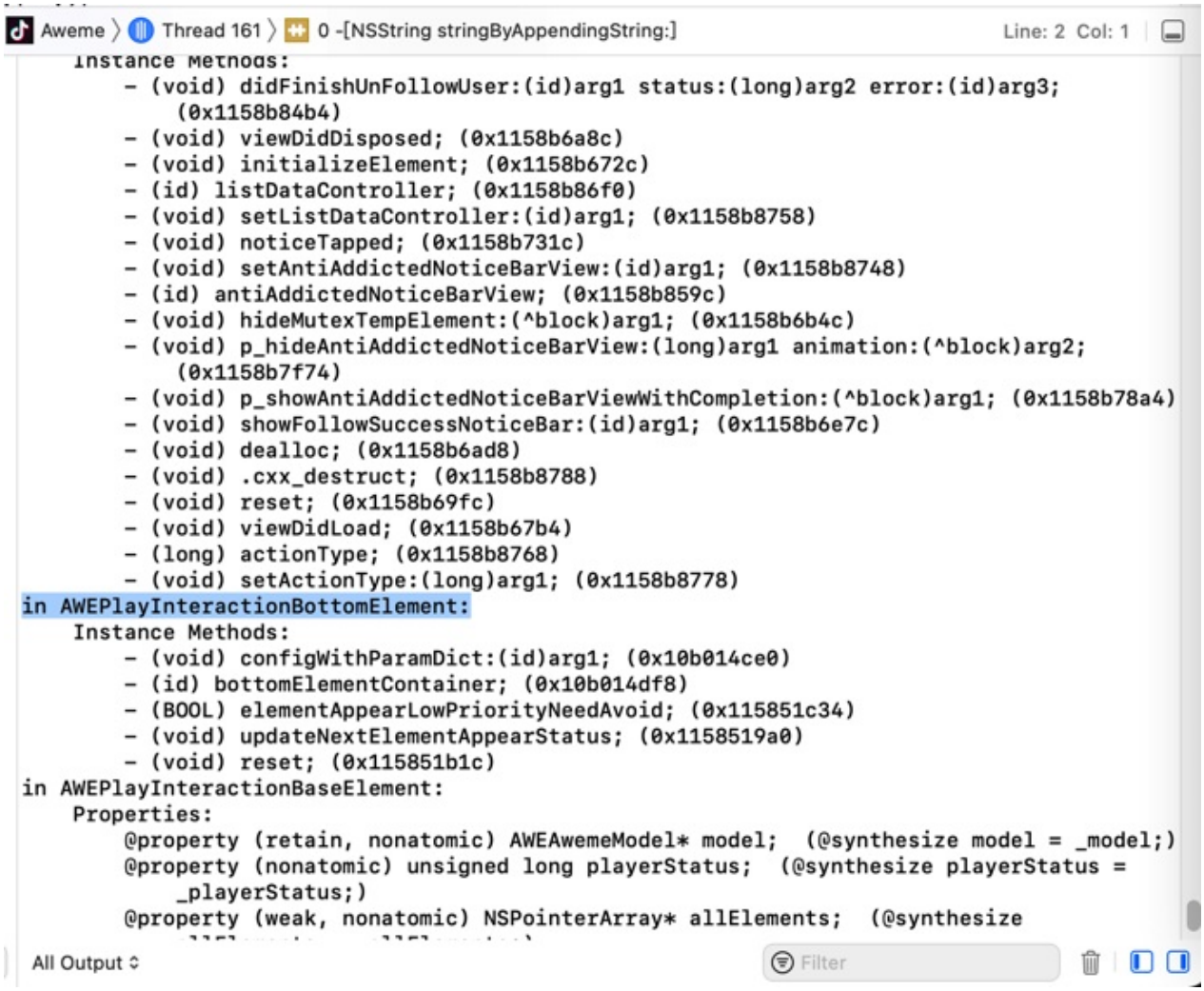
```
(lldb) po [objc_getClass("AWEPlayInteractionFollowSuccessElement") _shortMethodDescription]
2022-04-02 13:36:53.128548+0800 Aweme[45939:3543378] hook_misc.xm NSBundle$bundlePath: origBund
lePath: /usr/lib
<AWEPlayInteractionFollowSuccessElement: 0x10629ab90>:
in AWEPlayInteractionFollowSuccessElement:
    Properties
        @property (retain, nonatomic) AWEAntiAddictedNoticeBarView *antiAddictedNoticeBarView;
(@synthesize antiAddictedNoticeBarView = _antiAddictedNoticeBarView;)
        @property (retain, nonatomic) AWEHistoryPublicDataController *listDataController; (@sy
nthesize listDataController = _listDataController;)
        @property (nonatomic) long actionType; (@synthesize actionType = _actionType;)
        @property (readonly) unsigned long hash;
        @property (readonly) Class superclass;
```

```
        @property (readonly, copy) NSString * description;
        @property (readonly, copy) NSString * debugDescription;
    Instance Methods:
        - (void) didFinishUnFollowUser:(id)arg1 status:(long)arg2 error:(id)arg3; (0x1158b84b4)
        - (void) viewDidDisposed; (0x1158b6a8c)
        - (void) initializeElement; (0x1158b672c)
        - (id) listDataController; (0x1158b86f0)
        - (void) setListDataController:(id)arg1; (0x1158b8758)
        - (void) noticeTapped; (0x1158b731c)
        - (void) setAntiAddictedNoticeBarView:(id)arg1; (0x1158b8748)
        - (id) antiAddictedNoticeBarView; (0x1158b859c)
        - (void) hideMutexTempElement:(block)arg1; (0x1158b6b4c)
        - (void) p_hideAntiAddictedNoticeBarView:(long)arg1 animation:(block)arg2; (0x1158b7f74
)
        - (void) p_showAntiAddictedNoticeBarViewWithCompletion:(block)arg1; (0x1158b78a4)
        - (void) showFollowSuccessNoticeBar:(id)arg1; (0x1158b6e7c)
        - (void) dealloc; (0x1158b6ad8)
        - (void) .cxx_destruct; (0x1158b8788)
        - (void) reset; (0x1158b69fc)
        - (void) viewDidLoad; (0x1158b67b4)
        - (long) actionType; (0x1158b8768)
        - (void) setActionType:(long)arg1; (0x1158b8778)
in AWEPlayInteractionBottomElement:
    Instance Methods:
        - (void) configWithParamDict:(id)arg1; (0x10b014ce0)
        - (id) bottomElementContainer; (0x10b014df8)
        - (BOOL) elementAppearLowPriorityNeedAvoid; (0x115851c34)
        - (void) updateNextElementAppearStatus; (0x1158519a0)
        - (void) reset; (0x115851b1c)
in AWEPlayInteractionBaseElement:
    Properties
        @property (retain, nonatomic) AWEAwemeModel * model;  (@synthesize model = _model;)
        @property (nonatomic) unsigned long playerStatus;  (@synthesize playerStatus = _playerS
tatus;)
        @property (weak, nonatomic) NSPointerArray * allElements;  (@synthesize allElements = _a
llElements;)
        @property (readonly) unsigned long hash;
        @property (readonly) Class superclass;
        @property (readonly, copy) NSString * description;
        @property (readonly, copy) NSString * debugDescription;
    Instance Methods:
        - (struct CGRect) viewFrame; (0x11584af68)
        - (void) videoDidActivity; (0x10b0937f8)
        - (BOOL) alertIfNotValidForAction:(long)arg1; (0x11584aae8)
        - (id) elementFromAll:(id)arg1; (0x10b043ad8)
        - (void) viewController_viewWillDisappear; (0x11584abc4)
        - (void) viewController_viewDidDisappear; (0x11584abc8)
        - (void) viewController_didEndDisplaying; (0x11584abcc)
        - (void) viewController_willDisplay; (0x11584abb4)
        - (void) viewController_viewWillAppear; (0x11584abbc)
        - (void) viewController_viewDidAppear; (0x11584abc0)
        - (void) hideAllElementExcepts:(id)arg1; (0x11584a90c)
        - (void) updateAllElement; (0x11584ab74)
        - (void) setAllElements:(id)arg1; (0x10b019980)
        - (id) currentInfoForUnitWithIdentifier:(id)arg1; (0x11584aed8)
        - (void) hideProgressSliderPopView; (0x11584ac74)
```

```
        - (id) currentInfoForSubUnits; (0x11584adcc)
        - (id) currentInfoForUnitWithClassName (id)arg1; (0x11584ae5c)
        - (void) dealloc; (0x11584abe4)
        - (void) .cxx_destruct; (0x11584b090)
        - (void) pause; (0x11584ac54)
        - (void) resume; (0x11584ac64)
        - (void) setData (id)arg1; (0x10b04aecc)
        - (id) context; (0x10b00ef84)
        - (void) reset; (0x11584abd0)
        - (id) model; (0x10b016d44)
        - (void) setModel (id)arg1; (0x10b01cf58)
        - (void) play; (0x11584abe0)
        - (void) prepareForDisplay; (0x10b06b068)
        - (BOOL) isShowing; (0x11584b074)
        - (void) didEndDisplaying; (0x11584abb8)
        - (id) currentInfo; (0x11584acc0)
        - (unsigned long) playerStatus; (0x11584b080)
        - (void) setHide (BOOL)arg1; (0x10b0899c0)
        - (void) setPlayerStatus (unsigned long)arg1; (0x10b0431dc)
        - (id) allElements; (0x10b043c60)
in AWEBaseElement:
    Properties
        @property (weak, nonatomic) AWEElementContainer* elementContainer; (@dynamic elementCo
ntainer;)
        @property (weak, nonatomic) UIView* boxView; (@synthesize boxView = _boxView;)
        @property (weak, nonatomic) UIView* elementView; (@synthesize elementView = _elementVi
ew;)
        @property (nonatomic) BOOL hasCreateView; (@synthesize hasCreateView = _hasCreateView;)

        @property (retain, nonatomic) AWEPageContext* context; (@synthesize context = _context;
)
        @property (weak, nonatomic) AWEElementContainer* elementContainer; (@synthesize elemen
tContainer = _elementContainer;)
        @property (retain, nonatomic) UIView* view; (@synthesize view = _view;)
        @property (retain, nonatomic) id data; (@synthesize data = _data;)
        @property (readonly, nonatomic, getter isViewLoaded) BOOL viewLoaded;
        @property (copy, nonatomic) NSString* identity; (@synthesize identity = _identity;)
        @property (nonatomic) BOOL appear; (@synthesize appear = _appear;)
        @property (readonly, weak, nonatomic) UIViewController* viewController; (@synthesize v
iewController = _viewController;)
        @property (readonly) unsigned long hash;
        @property (readonly) Class superclass;
        @property (readonly, copy) NSString* description;
        @property (readonly, copy) NSString* debugDescription;
    Instance Methods:
        - (void) configWithParamDict (id)arg1; (0x11232b734)
        - (void) viewDidDisposed; (0x11232b730)
        - (void) processAppear (BOOL)arg1; (0x10b010138)
        - (id) elementContainer; (0x10b01036c)
        - (void) setAppear (BOOL)arg1; (0x10b01d350)
        - (void) initializeElement; (0x10b00f048)
        - (void) setElementContainer (id)arg1; (0x10b00ed6c)
        - (id) boxView; (0x11232b8fc)
        - (void) setBoxView (id)arg1; (0x11232b914)
        - (id) elementView; (0x11232b920)
        - (void) setElementView (id)arg1; (0x11232b938)
```

```
        - (BOOL) hasCreateView; (0x11232b944)
        - (void) setHasCreateView (BOOL)arg1; (0x11232b94c)
        - (void) addSubviewWithLayout (id)arg1 withEdgeInsets:(struct UIEdgeInsets)arg2; (0x112
32b9ec)
        - (void) addSubviewWithLayout (id)arg1 withEdgeInsets:(struct UIEdgeInsets)arg2 withHei
ght (double)arg3; (0x11232b9fc)
        - (void) hide (BOOL)arg1 duration:(double)arg2 animations (^block)arg3; (0x11232bcc4)
        - (void) hide (BOOL)arg1 duration:(double)arg2 withTransform (struct CGAffineTransform)
arg3 animations (^block)arg4; (0x11232bd48)
        - (void) addSubviewWithLayout (id)arg1; (0x11232b9d0)
        - (void) hide (BOOL)arg1 duration:(double)arg2; (0x11232bcb4)
        - (void) .cxx_destruct; (0x11232b968)
        - (id) data; (0x11232b8e8)
        - (void) setData (id)arg1; (0x11232b6cc)
        - (id) context; (0x11232b954)
        - (id) identity; (0x11232b8f0)
        - (void) setContext:(id)arg1; (0x11232b95c)
        - (void) setIdentity:(id)arg1; (0x10b014df0)
        - (id) view; (0x10b0101fc)
        - (void) setView (id)arg1; (0x11232b8dc)
        - (void) loadView; (0x10b0102e0)
        - (void) viewDidLoad; (0x11232b72c)
        - (BOOL) isViewLoaded; (0x10b0101ec)
        - (id) viewController; (0x10b012c08)
        - (BOOL) appear; (0x10b00f010)
(NSObject ...)
```


Aweme 〉 Thread 161 〉 0 -[NSString stringByAppendingString:]
Line: 2 Col: 1
(lldb) po [objc_getClass("AWEPlayInteractionFollowSuccessElement")
_shortMethodDescription]
2022-04-02 13:36:53.128548+0800 Aweme[45939:3543378] hook_misc.xm NSBundle$bundlePath:
origBundlePath=/usr/lib
<AWEPlayInteractionFollowSuccessElement: 0x10629ab90>:
in AWEPlayInteractionFollowSuccessElement:
Properties:
@property (retain, nonatomic) AWEAntiAddictedNoticeBarView*
antiAddictedNoticeBarView; (@synthesize antiAddictedNoticeBarView =
_antiAddictedNoticeBarView;)
@property (retain, nonatomic) AWEHistoryPublicDataController* listDataController;
(@synthesize listDataController = _listDataController;)
@property (nonatomic) long actionType; (@synthesize actionType = _actionType;)
@property (readonly) unsigned long hash;
@property (readonly) Class superclass;
@property (readonly, copy) NSString* description;
@property (readonly, copy) NSString* debugDescription;
Instance Methods:
- (void) didFinishUnFollowUser:(id)arg1 status:(long)arg2 error:(id)arg3;
(0x1158b84b4)
- (void) viewDidDisposed; (0x1158b6a8c)
- (void) initializeElement; (0x1158b672c)
- (id) listDataController; (0x1158b86f0)
- (void) setListDataController:(id)arg1; (0x1158b8758)
- (void) noticeTapped; (0x1158b731c)
- (void) setAntiAddictedNoticeBarView:(id)arg1; (0x1158b8748)
- (id) antiAddictedNoticeBarView; (0x1158b859c)
- (void) hideMutexTempElement:(^block)arg1; (0x1158b6b4c)
- (void) p_hideAntiAddictedNoticeBarView:(long)arg1 animation:(^block)arg2;
(0x1158b7f74)
- (void) p_showAntiAddictedNoticeBarViewWithCompletion:(^block)arg1; (0x1158b78a4)
- (void) showFollowSuccessNoticeBar:(id)arg1; (0x1158b6e7c)
- (void) dealloc; (0x1158b6ad8)
- (void) .cxx_destruct; (0x1158b8788)
- (void) reset; (0x1158b69fc)
- (void) viewDidLoad; (0x1158b67b4)
All Output
Filter



子教程

iOS逆向的动态调试中，已把部分独立内容整理到子教程：

- MonkeyDev
  - iOS逆向开发：MonkeyDev调试 (crifan.org)

相关代码：

- iOSYouTubeAdsFilter
  - crifan/iOSYouTubeAdsFilter: MonkeyDev+Xcode项目，iOS逆向YouTube，尝试实现广告过滤功能

附录

下面列出相关参考资料。

参考资料

- 【已解决】Cycript中找点击按钮的响应函数处理函数
- 【已解决】iPhone8中Cycript注入设置Preferences卡死
- 【已解决】cycript中如何打印窗口视图的详情包括字符串
- 【已解决】Cycript显示的页面元素不是最新的当前弹框页面
- 【已解决】iOS的ObjC中如何获取到最顶层的窗口视图
- 【已解决】iOS逆向：palera1n越狱后iPhone中初始化安装Cycript环境
- 【已解决】iOS逆向Apple账号：搭建Cycript调试系统设置UI界面
- 【已解决】Mac中下载和安装Reveal
- 【已解决】MonkeyDev中如何使用Reveal调试YouTube广告页面元素
- 【已解决】Mac中Reveal中看不到MonkeyDev调试的iPhone设备
-
- 【记录】XCode+MonkeyDev动态调试YouTube类：YTWatchMiniBarViewController
-
- iOS逆向攻防实战 - 掘金 (juejin.cn)
- SpringBoard tweak 双击图标启动debugserver - 干货分享 - 睿论坛
- Frida & Passionfruit 安装记录. Frida是一个功能强大且可扩展的工具包，具有众多优势，非常适合测试和评估And… | by iOS Jailbreak Notes | Medium
- Reveal 24 (12917) 破解版 for Mac iOS界面UI开发调试神器 (macwk.com)下载到
- Reveal*24*(12917)__macwk.com.dmg__macwk.com.dmg)
- Cycript Tricks - iPhone Development Wiki
- iOS攻防（六）：使用Cycript一窥运行程序的神秘面纱(入门篇) | 曹雪松de博客|CoderBoy's Blog (sevencho.github.io)
- iOS逆向之Cycript的使用 - 简书 (jianshu.com)
- Cycript的一些基础使用 - 简书 (jianshu.com)
- CoderMJLee/mjcript: 【越狱-逆向】基于Cycript实现的一些实用函数 (github.com)
- mjcript/mjcript.cy at master · CoderMJLee/mjcript (github.com)
-