
目录

前言	1.1
iOS底层机制概览	1.2
Apple开发资料	1.3
iOS逆向常涉及内容	1.3.1
底层机制逆向心得	1.4
附录	1.5
参考资料	1.5.1

iOS逆向开发：iOS底层机制

- 最新版本: v0.6
- 更新时间: 20221024

简介

介绍iOS逆向开发期间涉及到iOS底层机制方面的，主要是ObjC方面的内容。包括Block回调、ObjC的Runtime运行时、Apple苹果相关开发资料。以及一些逆向的动态调试中涉及到iOS底层机制的相关心得。

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_ios_internal: iOS逆向开发：iOS底层机制](#)

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：iOS底层机制 book.crifan.org](#)
- [iOS逆向开发：iOS底层机制 crifan.github.io](#)

离线下载阅读

- [iOS逆向开发：iOS底层机制 PDF](#)
- [iOS逆向开发：iOS底层机制 ePUB](#)
- [iOS逆向开发：iOS底层机制 MOBI](#)

版权和用途说明

此电子书教程的全部内容，如无特别说明，均为本人原创。其中部分内容参考自网络，均已备注了出处。如有版权，请通过邮箱联系我 [admin 艾特 crifan.com](mailto:admin@crifan.com)，我会尽快删除。谢谢合作。

各种技术类教程，仅作为学习和研究使用。请勿用于任何非法用途。如有非法用途，均与本人无关。

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。

更多其他电子书

本人 [crifan](#) 还写了其他 100+ 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](#)

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2022-10-24 16:12:09

iOS底层机制概览

iOS逆向中，尤其是[动态调试](#)期间，往往涉及iOS的，尤其是ObjC的，的底层机制和实现原理。

此处整理相关的内容：

- ObjC
 - Block
 - 详见独立教程：
 - 【整理Book】iOS逆向开发：Block的相关基础知识
 - Runtime运行时
 - 详见独立教程：
 - 【整理Book】iOS逆向开发：ObjC运行时Runtime
- Apple苹果
 - 相关开发资料

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2022-10-24 14:55:37

Apple苹果相关开发资料

TODO:

- 【已解决】iOS中st_size的off_t是什么类型
 - 【已解决】iOS或Linux或C中pid_t的定义
-

此处整理，Apple苹果的，和iOS逆向相关的，尤其是涉及到iOS底层机制方面的，开发资料。

- Apple = 苹果
 - 相关开发资料
 - 源码 源代码
 - 官网文档

Apple苹果官网

源码+文档

概述=总入口：

[Open Source - Apple Developer](#)

详解：

- 开源代码Open Source Projects
 - Apple Open Source
 - <https://opensource.apple.com>
 - 文档
 - Kernel
 - <https://developer.apple.com/library/mac/#documentation/Darwin/Conceptual/KernelProgramming/build/build.html>
 - Frameworks
 - https://developer.apple.com/library/mac/#documentation/MacOSX/Conceptual/OSX_Technology_Overview/SystemFrameworks/SystemFrameworks.html
 - Security
 - https://developer.apple.com/library/archive/documentation/Security/Conceptual/Security_Overview/Introduction/Introduction.html
 - 源码
 - 离线下载 源码 总入口
 - <https://opensource.apple.com/tarballs/>
 - 子模块
 - ObjC Runtime
 - <https://opensource.apple.com/tarballs/objc4/>
 - xnu
 - <https://opensource.apple.com/tarballs/xnu/>
 - dyld
 - <https://opensource.apple.com/tarballs/dyld/>
 - cctools
 - <https://opensource.apple.com/tarballs/cctools/>
 - 在线浏览 源码 总入口
 - <https://opensource.apple.com/source/>
 - 子模块

- xnu
 - <https://opensource.apple.com/source/xnu/>
 - kern
 - https://opensource.apple.com/source/xnu/xnu-792bsd/kern/kern_sysctl.c.auto.html
 - dlyd
 - <https://opensource.apple.com/source/dyld/>
 - system_cmds
 - https://opensource.apple.com/source/system_cmds/
 - sysctl
 - system_cmds-880.60.2
 - https://opensource.apple.com/source/system_cmds/system_cmds-880.60.2/sysctl.tproj/
 - sysctl的源码
 - https://opensource.apple.com/source/system_cmds/system_cmds-880.60.2/sysctl.tproj/sysctl.c.auto.html
 - Libc
 - posix_spawn
 - https://opensource.apple.com/source/Libc/Libc-825.25/sys posix_spawn.c.auto.html
 - objc4 = Objc
 - <https://opensource.apple.com/source/objc4/>
 - 不同版本
 - <https://opensource.apple.com/source/objc4/objc4-818.2/>
 - <https://opensource.apple.com/source/objc4/objc4-532>
 - <https://opensource.apple.com/source/objc4/objc4-646>
 - 子模块
 - runtime
 - [objc_release](https://opensource.apple.com/source/objc4/objc4-532/runtime/NSObject.mm.auto.html)
 - <https://opensource.apple.com/source/objc4/objc4-532/runtime/NSObject.mm.auto.html>
 - [objc_alloc](https://opensource.apple.com/source/objc4/objc4-646/runtime/objc-internal.h)
 - <https://opensource.apple.com/source/objc4/objc4-646/runtime/objc-internal.h>
 - [objc_msgSendSuper2](https://opensource.apple.com/source/objc4/objc4-532/runtime/objc-abi.h.auto.html)
 - <https://opensource.apple.com/source/objc4/objc4-532/runtime/objc-abi.h.auto.html>
 - [objc_retainBlock](https://opensource.apple.com/source/objc4/objc4-493.9/runtime/objc-arr.mm.auto.html)
 - <https://opensource.apple.com/source/objc4/objc4-493.9/runtime/objc-arr.mm.auto.html>
 - libpthread
 - [pthread_get_stackaddr_np](https://opensource.apple.com/source/libpthread/libpthread-105.10.1/src/pthread.c.auto.html)
 - <https://opensource.apple.com/source/libpthread/libpthread-105.10.1/src/pthread.c.auto.html>
 - MacOS Forge
 - www.macosforge.org

函数和命令的文档

注: `man` = `manual` = 手册

- Apple文档总入口: [Documentation Archive \(apple.com\)](http://Documentation Archive (apple.com))
 - 函数 `man` 手册 总入口: [API Reference: iOS Manual Pages \(apple.com\)](http://API Reference: iOS Manual Pages (apple.com))

- 分很多大类
 - Section 2: system calls and error numbers
 - Section 2 of the manual contains documentation on UNIX system calls, error codes, and C library routines that wrap system calls. Most of these functions are described in headers that reside in /usr/include/sys.
 - For a detailed introduction, see intro(2)
 - [Mac OS X Manual Page For intro\(2\) \(apple.com\)](#)
 - Section 3: C libraries
 - Section 3 of the manual contains documentation on C library routines. This section excludes library routines that merely wrap UNIX system calls. Most of these functions are described in headers that reside in /usr/include or subdirectories therein.
 - For a detailed introduction, see intro(3)
 - [Mac OS X Manual Page For intro\(3\) \(apple.com\)](#)
 - Section 3cc: 加密 解密 算法 相关
 - [API Reference: iOS Manual Pages \(apple.com\)](#)
 - Section 3ssl
 - Section 3ssl of the manual contains documentation on OpenSSL library routines. These functions are described in headers that reside in /usr/include/openssl, and are split between the libssl and libcrypto libraries
 - For a detailed introduction, see crypto(3) and ssl(3)
 - [Mac OS X Manual Page For crypto\(3ssl\) \(apple.com\)](#)
 - crypto - OpenSSL cryptographic library
 - [Mac OS X Manual Page For ssl\(3ssl\) \(apple.com\)](#)
 - SSL - OpenSSL SSL/TLS library
 - Section 3x
 - Section 3x of the manual contains documentation on curses-related library routines used for general formatting on text terminals. These functions are described in headers that reside in /usr/include, and are located in the libncurses library.
 - For a detailed introduction, see ncurses(3)
 - [Mac OS X Manual Page For ncurses\(3x\) \(apple.com\)](#)
 - ncurses - CRT screen handling and optimization package
 - Section 5
 - Section 5 of the manual contains documentation on file formats and conventions. It includes documentation about file-system data structures, information about configuration files for various daemons, and information about data structures used in various binary and text file formats used by various parts of the operating system.
 - For a detailed introduction, see intro(5).
 - [Mac OS X Manual Page For manpages\(5\) \(apple.com\)](#)
 - manpages -- An introduction to manual pages

log日志的字符串格式化参数语法

[String Format Specifiers \(apple.com\)](#)

The iPhone Wiki

syscall

[Kernel Syscalls - The iPhone Wiki](#)

hw.machine 机型 映射

- [Models - The iPhone Wiki](#)

其他来源

内核原理和机制

[OS Internals: \(newosxbook.com\)](#)

->

- [MAC OS X Internals: A Systems Approach: Singh, Amit: 9780321278548: Books: Amazon.com](#)
- [J's Entitlement DataBase \(newosxbook.com\)](#)
 - OS X/iOS Entitlement Database - v0.8
 - As compiled by Jonathan Levin, @Morpheus_
 - Now with entitlements from iOS 9.0.2 through 15.2, MacOS 11.4 through 15.3

iOS逆向英文教程

据说是第一本英文书的专门详细介绍iOS逆向的书：

[iosre/iOSAppReverseEngineering: The world's 1st book of very detailed iOS App reverse engineering skills :\) \(github.com\)](#)

->

<https://github.com/iosre/iOSAppReverseEngineering/blob/master/iOSAppReverseEngineering.pdf>

有需要可以学习和参考。

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2022-10-24 16:09:13

iOS逆向常涉及内容

在iOS逆向期间，常涉及到很多Apple苹果相关的开发资料，整理如下供参考。

iOS Runtime Header

可以查询和搜索iOS运行时的头文件的网站：

<https://developer.limneos.net/>

ObjC的类

UIDevice

[UIDevice | Apple Developer Documentation](#)

ObjC Runtime

- objc4 = Objc = ObjC Runtime
 - 下载
 - <https://opensource.apple.com/tarballs/objc4/>
 - 在线浏览
 - <https://opensource.apple.com/source/objc4/>
 - 常用版本
 - <https://opensource.apple.com/source/objc4/objc4-750/>
 - <https://opensource.apple.com/source/objc4/objc4-818.2/>

具体例子：

- objc_retainAutoreleasedReturnValue + objc_autoreleaseReturnValue
 - <https://opensource.apple.com/source/objc4/objc4-532.2/runtime/NSObject.mm>
- objc_retainBlock
 - 相关源码: runtime/objc-arr.mm
 - 离线下载
 - <https://opensource.apple.com/tarballs/objc4/objc4-493.9.tar.gz>
 - 在线浏览
 - <https://opensource.apple.com/source/objc4/objc4-493.9/runtime/objc-arr.mm.auto.html>
- objc_alloc
 - <https://opensource.apple.com/source/objc4/objc4-646/runtime/objc-internal.h>

```
OBJC_EXPORT id objc_alloc(Class cls)
_OBJC_AVAILABILITY_STARTING(__MAC_10_9, __IPHONE_7_0);

OBJC_EXPORT id objc_allocWithZone(Class cls)
_OBJC_AVAILABILITY_STARTING(__MAC_10_9, __IPHONE_7_0);
```

- objc_loadWeakRetained
 - <https://opensource.apple.com/source/objc4/objc4-706/runtime/NSObject.mm.auto.html>

XNU

由于 xnu 很重要，iOS逆向期间经常涉及到，所以单独介绍相关内容：

- xnu
 - 离线下载
 - <https://opensource.apple.com/tarballs/xnu/>
 - 在线浏览
 - <https://opensource.apple.com/source/xnu/>

查看自己的xnu版本

对于自己的越狱手机，此处的iPhone7，去查看对应的xnu的版本：

```
→ ~ ssh root@192.168.0.33
iPhone7:~ root# uname -a
Darwin iPhone7 19.6.0 Darwin Kernel Version 19.6.0: Sat Jun 27 04:35:37 PDT 2020; root:xnu-6153.142.1~4/RELEASE_ARM64_T8010 iPh
one9,1 arm64 D10AP Darwin
```

此处被测的已越狱的iPhone的xnu是：

xnu-6153.142.1

去官网找对应版本的代码：

[xnu Source Browser \(apple.com\)](#)

没看到这个版本

-> 只能找到，最接近的版本：

- xnu-6153.141.1.tar.gz
 - <https://opensource.apple.com/tarballs/xnu/xnu-6153.141.1.tar.gz>

可下载下来，供后续参考研究。

iOS中的基本类型的定义

关于iOS中的很多相关的底层的类型：

- `_darwin_mode_t`
- `_darwin_off_t`
- `_darwin_pid_t`

的定义是：

```
typedef __uint16_t    __darwin_mode_t;           /* [???] Some file attributes */
typedef __int64_t     __darwin_off_t;            /* [???] Used for file sizes */
typedef __int32_t     __darwin_pid_t;             /* [???] process and group IDs */
```

来源：

- Apple的opensource
 - https://opensource.apple.com/source/xnu/xnu-792/bsd/sys/_types.h
- 其他
 - [apple/darwin-xnu: The Darwin Kernel \(mirror\)](#)
 - https://github.com/apple/darwin-xnu/blob/main/bsd/sys/_types.h

头文件 errno.h

- |旧版本： xnu-201
 - <https://opensource.apple.com/source/xnu/xnu-201/bsd/sys/errno.h>

```
#define ENOTSUP 45           /* Operation not supported */
#ifndef _POSIX_SOURCE
#define EOPNOTSUPP ENOTSUP    /* Operation not supported */
```

- 新版本: xnu-792
 - <https://opensource.apple.com/source/xnu/xnu-792/bsd/sys/errno.h.auto.html>

```
#define ENOTSUP 45           /* Operation not supported */
```

结论:

- ENOTSUP = 45

在线浏览 xnu 代码

[XXR - XNU cross reference - Alpha \(newosxbook.com\)](#)

The screenshot shows the XXR - The XNU Xref interface. At the top, it says "XXR – The XNU Xref" and "(a Version 0.4 (with XNU-8019 from Apple GitHub and falk Skyyw's indexing) feedback appreciated". Below that, there are dropdown menus for "Active Tree" set to "xnu-8019 (MacOS 17, Darwin 21 -)" and "Search for" with a search bar. The main area is a file browser for "xnu-8019/". On the left, there's a sidebar with links like "upstream_base_commits", "APPLE LICENSE", "EXTERNAL_HEADERS/", "Makefile", "README.md", "SETUP/", "bsd/", "config/", "doc/", "iokit/", "libkdd/", "libkern/", "libsa/", "libsyscall/", "makedefs/", "osfmk/", "pexpert/", "san/", "security/", "tests/", and "tools/". A tooltip over the sidebar says "This is xnu-8019. See this file in xnu-8019 (MacOS 17.0, Darwin 21 -)". To the right of the sidebar is a list of other XNU versions: xnu-6153.11.26 (MacOS 15.0, Darwin 19 -), xnu-4903.221.2 (MacOS 14.0, Darwin 18), xnu-4570.41.2 (MacOS 13.3, Darwin 17), xnu-3789.70.16 (MacOS 12.6, Darwin 16), xnu-3248.20.55 (MacOS 11.2, Darwin 15), xnu-2782.40.9 (MacOS 10.10.5, Darwin 14), xnu-2422.115.4 (MacOS 10.9.5, Darwin 13), xnu-2050.24.15 (MacOS 10.8.4, Darwin 12), xnu-1699.32.7 (MacOS 10.7.5, Darwin 11), xnu-1504.15.3 (MacOS 10.6.8, Darwin 10), and xnu-1228.15.4 (MacOS 10.5.8, Darwin 9).

man手册文档

前面介绍的

Apple 函数 man 手册 总入口: [API Reference: iOS Manual Pages \(apple.com\)](#)

中有很多, iOS逆向期间, 常常会涉及到的一些 API 或命令, 整理如下:

- Section 2: system calls and error numbers
 - ENOMEM 错误码定义
 - 12 = ENOMEM Cannot allocate memory
 - The new process image required more memory than was allowed by the hardware or by system-imposed memory management constraints. A lack of swap space is normally temporary; however, a lack of core is not. Soft limits may be increased to their corresponding hard limits.
 - stat64
 - [Mac OS X Manual Page For stat64\(2\)](#)

- [Section 3: C libraries](#)
 - [system](#)
 - [Mac OS X Manual Page For system\(3\)](#)
 - [sysctl](#)
 - [Mac OS X Manual Page For sysctl\(3\)](#)
 - [strlen](#)
 - [Mac OS X Manual Page For strlen\(3\)](#)

iPhone iOS SDK 源码

- [iPhoneOS13.0.sdk](#)
 - [iOS-SDKs/iPhoneOS13.0.sdk at master · xybp888/iOS-SDKs \(github.com\)](#)
 - [stat.h](#)
 - [iOS-SDKs/stat.h at master · xybp888/iOS-SDKs \(github.com\)](#)
 - [syscall.h](#)
 - [iOS-SDKs/syscall.h at master · xybp888/iOS-SDKs \(github.com\)](#)
 - [sysctl.h](#)
 - [iOS-SDKs/sysctl.h at master · xybp888/iOS-SDKs \(github.com\)](#)
 - [types.h](#)
 - [iOS-SDKs/types.h at master · xybp888/iOS-SDKs \(github.com\)](#)

改机相关

sysctl相关

- 苹果官网文档
 - [sysctlbyname](#)
 - [sysctlbyname | Apple Developer Documentation](#)
 - [sysctl](#)
 - [sysctl | Apple Developer Documentation](#)
- man手册
 - [SYSCTL](#)
 - [Mac OS X Manual Page For sysctlbyname\(3\) \(apple.com\)](#)
- 源码
 - [sysctl.c \(apple.com\)](#)
- 其他
 - [sysctlbyname \(freebsd.org\)](#)
 - [sysctlbyname\(3\) manual page \(lemoda.net\)](#)
 - [sysctl, sysctlbyname \(qnx.com\)](#)
 - [sysctlbyname.3 \(daemon-systems.org\)](#)

其他

iOS中 属性列表 Property List = plist

- [Introduction to Property Lists \(apple.com\)](#)

C语言相关开发整理和心得

gcc

编译时-Wxxx的参数：

- [Warning Options \(Using the GNU Compiler Collection \(GCC\)\)](#)

clang

编译时参数：

- [Clang Compiler User's Manual — Clang 13 documentation \(llvm.org\)](#)
- [Clang command line argument reference — Clang 13 documentation \(llvm.org\)](#)

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新：2022-10-24 16:07:25

iOS底层机制逆向心得

此处整理，iOS逆向时，涉及到iOS的底层机制方面的，心得和经验总结。

通用

- 【整理】编程基础知识：函数Prologue开场白和函数Epilogue结尾
- 【整理】iOS逆向心得：给函数加了hook同时加断点会导致EXC_BREAKPOINT的崩溃
- 【已解决】iOS逆向心得：类没有setXxx函数但是有属性xxx
- 【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
- 【已解决】iOS中的caddr_t类型的定义
 - TODO：
 - 把各种iOS逆向期间，涉及的各种类型的定义，也整理过来

动态调试

- 【已解决】iOS逆向心得：如何从对x8的adrp和ldr计算出对应的qword字符串值
- 【整理】iOS逆向调试心得：bool等变量类型
- 【整理】iOS逆向心得：变量类型是bool类型

po

- 【整理】iOS逆向心得：当iPhone锁屏时Xcode中lldb的po会卡死
- 【整理】iOS逆向心得：po异常时NSString的字符串无法像char*一样打印出来
- 【整理】iOS逆向调试心得：po不是对象实例但可以看到是哪个类

objc_msgSend

- 【未解决】IDA中如何解析objc_msgSend函数调用
- 【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

类

- 【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
- 【整理】iOS逆向心得：打印ObjC类的属性
- 【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逆向调试心得：给类的属性去设置值以及如何计算类的属性的偏移量
- 【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

函数

- 【整理】iOS逆向调试心得：bl函数调用和返回常见逻辑
- 【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lldb调试心得：iOS的ObjC的无名汇编跳板函数
 - 相关

- 【已解决】clang中的__cdecl和支持哪些调用规范
- 【已解决】微软的调用规范的参数传递和命名规范
- 【已解决】iOS中调用asm汇编关键字：asm asm asm和volatile volatile__
- 【已解决】XCode的断点条件判断中如何获取iOS的ObjC函数的参数值

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2022-10-24 14:39:00

附录

下面列出相关参考资料。

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2022-03-17 20:39:28

参考资料

- OS Internals: (newosxbook.com)
- MAC OS X Internals: A Systems Approach: Singh, Amit: 9780321278548: Books: Amazon.com
- XXR - XNU cross reference - Alpha (newosxbook.com)
- Open Source - Apple Developer
- Documentation Archive (apple.com)
- Mac OS X Manual Page For stat64(2) (apple.com)
- Kernel Syscalls - The iPhone Wiki
- 越狱检测抖音逻辑??? - Cydia | 微信公众号文章阅读 - WeMP
-

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2022-10-24 15:41:43