

码农13年
阅读量: 56 浏览量:

原创 1万+ 2617 246万+ 等级

创作 周排行 总排行 访问 等级

1万+	844	598	239	1077
积分	粉丝	获赞	评论	收藏

热门文章

- Unicode【真正的完整码表】对照表（一）
[@ 553891](#)
- Unicode【真正的完整码表】对照表（二）
关于Unicode码点 @ 431112
- 汉字排版的GBK编码（附完整码表）
[@ 76059](#)
- HTTPS协议详解(四)：TLS/SSL握手过程
[@ 60498](#)
- http状态码204理解 @ 68511

最新评论

iOS设备证书: Provisioning profile "xx_D...m0_58983241": EmployeeKey的"phoneNu...merical创建UNIQUE约束。

主线程 [__NSDiscontinuationObjectForKey]...herina 显的，由于是别的组的bug只是做了一个方案。这个方案并不是最好的。

TCP 滑动窗口协议比传输层窗口协议...weixin_52864440: 讲的真棒

Unicode【真正的完整码表】对照表（二...weixin_42548140: 赞

HTTPS协议详解(四)：TLS/SSL握手过程...sunny_corona: 没有无效的

最新文章

Found an unexpected Mach-O header code: 0x782613c21

主线程 [__NSDiscontinuationObjectForKey] SEGV_ACCERR 类型的orash

push/Egress 两种方式。
viewWillDisAppear、viewDidDisAppear的顺序

2021年	2020年
2月	7篇
5月	12篇
7月	31篇
10月	68篇
11月	

掘金

himeria2020-11-06 17:50:48973收藏

分类专栏Objective-C高级编程文章标签iOS

Objective-C高级编程

专栏收录该内容

12订阅19篇文章

订阅专栏

【转载】作者：孔猪碌谷

链接：<https://juejin.im/post/6844903688608153614>

用户在使用App的过程中，经常遇到闪退的情况，体验不太好，本文尝试探索引发闪退的原因，以及在遇到crash的情况下，尽可能的保持程序运行，并及时上报错误。

一、crash类型

1.OC层面的crash

1.1 普通类型

- NSInvalidArgumentException：非法参数异常，传入非法参数导致异常，nil参数比较常见。
- NSRangeException：下标越界导致的异常。
- NSGenericException：foreach的循环当中修改元素导致的异常。

1.2 KVO

KVO Crash常见原因：

- 移除未注册的观察者
- 重复移除观察者
- 添加了观察者但是没有实现-observeValueForKeyPath:ofObject:change:context:方法
- 添加移除keypath=nil
- 添加移除observer=nil

1.3 unrecognized selector sent to instance

对象接收到未知的消息,即下图中消息未能处理的情况。

```
image<figcaption></figcaption>
```

2.Signal层面的crash

除了OC层面的异常捕获之外，很多内存错误、访问错误的地址产生的crash则需要利用unix标准的signal机制，注册SIGABRT, SIGBUS, SIGSEGV等信号发生时的处理函数。该函数中我们可以输出栈信息，版本信息等其它一切我们所想要的。

- SIGKILL:用来立即结束程序的运行的信号。
- SIGSEGV:试图访问来分配给自己的内存，或试图往没有写权限的内存地址写数据。
- SIGABRT:调用abort函数生成的信号。
- SIGTRAP:由断点指令或其它trap指令产生。
- SIGBUS:非法地址，包括内存地址对齐(alignment)出错。比如访问一个四个字长的整数，但其地址不是4的倍数。它与SIGSEGV的区别在于后者是由于对合法存储地址的非法访问触发的(如访问不属于自己存储空间或只读存储空间)。

二、存在问题

- 程序闪退，用户体验不好

三、监听crash

一、crash 类型

1. OC 崩溃的 crash
 - 1.1 普通崩溃
 - 1.2 KVO
 - 1.3 unrecognized selector sent to...
2. Signal 崩溃的 crash

二、存在问题

三、监听 crash

1. 任程序段内退并上报
 - 1.1 NSSetUncaughtExceptionHandler...
 - 1.2 AppDelegate 中注册 SIGABRT...
2. Crash 自动修复+捕获上报

👉 4 行代码 30 分钟在 APP 中实现互动直播

每月 1 万分钟免费

立即试用

分类专题

- Objective-C 高级编程 18 篇
- 字符编码介绍 11 篇
- OC block 专题 7 篇
- 密码学—RSA 非对称加密 6 篇
- iOS 内存管理剖析 7 篇
- IM 软件业务知识 12 篇
- iOS push 开发专题 10 篇
- https/https 协议详解 6 篇

	混合	16.00
	商业类知识	10.00
	设计模式	
	OpenGL	1.00
	开源库	
	密码学/网络安全	12.00
	Objective-C	42.00
	数据库	3.00
	Xcode	10.00
	TCP、IP/网络编码	17.00
	XML/ProtocolBuf	4.00
	计算机系统	9.00
	算法	
	C/C++（数据结构）	10.00
	iOS平台知识	33.00
	iOS push通知	9.00
	iOS UI框架	21.00
	iOS SDK	18.00
	block编程	4.00
	图像	1.00
	XML/ProtocolBuf	1.00
	前端	2.00

```

9  signal(SIGSEGV, SignalExceptionHandler);
10 signal(SIGFPE, SignalExceptionHandler);
11 signal(SIGBUS, SignalExceptionHandler);
12 signal(SIGPIPE, SignalExceptionHandler);
13 }
14 复制代码

```

```

1  void SignalExceptionHandler(int signal)
2  {
3
4      NSMutableString *mstr = [NSMutableString alloc init];
5      [mstr appendString:@"Stack:\n"];
6      void* callstack[128];
7      int i, frames = backtrace(callstack, 128);
8      char** strs = backtrace_symbols(callstack, frames);
9      for (i = 0; i < frames; ++i) {
10         [mstr appendFormat:@"%s\n", strs[i]];
11     }
12     [SignalHandler saveCrash:mstr];
13 }
14 }
15 复制代码

```

2. Crash自动修复+捕获上报

2.1 针对普通类型Crash的处理机制

hook相关的代码，增加保护机制。以NSArray越界为例，hook objectAtIndex方法，在方法中捕获越界异常，并在最后返回一个nil对象。

```

1  [self exchangeInstanceMethod:___NSArrayI_method1Sel:@selector(objectAtIndex:) method2Sel:@selector(avoidCrashAtIndex:)]
2  复制代码

```

```

1  - (id)avoidCrashObjectAtIndex:(NSUInteger)index {
2      id object = nil;
3
4      @try {
5          object = [self avoidCrashObjectAtIndex:index];
6      }
7      @catch (NSException *exception) {
8          // 捕获异常，根据exception打印出堆栈信息，同时也避免了程序崩溃
9      }
10     @finally {
11         return object;
12     }
13 }
14 复制代码

```

注意：使用方法进行捕获异常之后，第三方工具将不会搜集到崩溃信息并上报，需要在catch中手动上报。

2.2 针对KVO Crash的处理机制

2.3 针对 unrecognized selector 解决方案

通常，当我们不能确定一个对象是否能接收某个消息时，会先用 `respondToSelector:` 来判断一下。如下代码所示：

```

1 if ([self respondsToSelector:[selector method]]) {
2     [self performSelector:[selector method]];
3 }
4 复制代码

```

当一个对象无法接收某一消息时，就会启动所谓“消息转发 (message forwarding)”机制。通过这一机制，我们可以告诉对象如何处理该消息。默认情况下，对对象接收到未知的消息，会导致程序崩溃。

image

<figcaption></figcaption>

上图可以看出，在一个函数找不到时，Objective-C 提供了三种方式去补救：

- 调用 `resolveInstanceMethod` 给机会让类添加这个实现这个函数
- 调用 `forwardingTargetForSelector` 让别的对象去执行这个函数
- 调用 `methodSignatureForSelector` (函数符号制造器) 和 `forwardInvocation` (函数执行器) 灵活的将目标函数以其他形式执行

如果都不中，调用 `doesNotRecognizeSelector` 抛出异常。

```

- (void)forwardInvocation: (NSInvocation *) anInvocation

{
    [self methodSignature: *) methodSignatureForSelector: (SEL) aSelector
}

```

方法一：hook 上述两个方法，在 `methodSignatureForSelector` 中返回有效的 `NSMethodSignature`，在 `forwardInvocation` 中添加 `try-catch` 即可，代码如下：

```

1 [self exchangeInstanceMethod:[self class] methodSel:[selector methodSignatureForSelector]: method2Sel:[selector]
2 [self exchangeInstanceMethod:[self class] methodSel:[selector forwardingInvocation]: method2Sel:[selector]
3 复制代码

```

```

1 - (NSMethodSignature *)avoidCrashMethodSignatureForSelector:(SEL)aSelector
2 {
3     NSMethodSignature *ms = [self avoidCrashMethodSignatureForSelector:aSelector];
4     if ([self respondsToSelector:aSelector] || ms){
5         return ms;
6     }
7     else{
8         return [SafeProxy instanceMethodSignatureForSelector:@selector(safe_crashLog)];
9     }
10 }
11
12 - (void)avoidCrashForwardInvocation:(NSInvocation *)anInvocation {
13
14     @try {
15         [self avoidCrashForwardInvocation:anInvocation];
16     }
17     @catch (NSException *exception) {
18         // 捕获异常，根据exception打印出堆栈信息，同时也避免了程序崩溃
19         // 上报
20     }
21     @finally {
22     }
23 }
24 复制代码

```

方法二：直接hook `doesNotRecognizeSelector` 也可实现，`doesNotRecognizeSelector` 起到抛出异常的作用，自己增加try-catch进行捕获即可，代码如下：

```

1 [self exchangeInstanceMethod:[self class] method1Sel:@selector(doesNotRecognizeSelector): method2Sel:@selector(safe_crashLog)];
2 复制代码

```

```

1 - (void)avoidCrashDoesNotRecognizeSelector:(SEL)aSelector{
2     @try {
3         [self avoidCrashDoesNotRecognizeSelector:aSelector];
4     }
5     @catch (NSException *exception) {
6         // 捕获异常，根据exception打印出堆栈信息，同时也避免了程序崩溃
7         // 上报
8     }
9     @finally {
10    }
11 }
12 复制代码

```

效果如下：

```
1  NSInvalidArgumentException
2  error = _dispatchSourceError initWithObjects:count:]: attempt to insert nil object from objects[i]
3  Error Place:[ViewController NSArray_Test_InstanceArray]
4  AvoidCrash default is to remove nil object and instance a array.
5  复制代码
```

打印出了堆栈信息，同时避免了程序崩溃。

注意：使用方法进行捕获异常之后，第三方工具将不会搜集到崩溃信息并上报，需要在catch中手动上报。

2.4 针对野指针的处理机制

模拟Xcode中的zombie机制：

- 1.Swizzle原有allocWithZone方法，添加野指针防护标记。
- 2.Swizzle原有dealloc方法，如果有野指针防护标记，调用objc_destructInstance方法，修改实例isa使其指向zombieObjc，保留原始类名，以便上报使用。
- 3.Swizzle消息转发机制forwardingTargetForSelector方法，处理所有原始类originObject的方法，收集错误信息并上报。
- 4.及时释放zombieObjc。

注：objc_destructInstance会释放与实例相关联的引用，但是并不释放该实例的内存。

华为技术有限公司 创新 实践 赋能 生态

© 2021 Huawei

为iOS开发安卓应用和安卓应用大迁移过程中 300万+读者等你来读， 免费全本试读， 打造全球最懂数字创新创业、 探索智慧未来！	
iOS 异常捕捉及发送处理，ExceptionHandler。	07
iOS异常捕捉发送，发送邮件及写入日志工具。提供前缀、错误的具体位置（类名及行号），所有异常报告内容，及异常报告文件路径地址，并统一。	
优秀评论可以帮你查看获得更高权重	
评论	
相关推荐	
iOS中常见Crash总结_weixin_33827731的博客	6
iOS中常见Crash总结 [TOC] 1、找不到方法的实现 unrecognized selector sent to instance 2、KVC造成的Crash 3、EXC_BAD_ACCESS 4、KVO引起的Crash	
iOS各种异常Crash总结_fishma的专栏	
iOS各种异常Crash总结 1、找不到方法的实现 unrecognized selector sent to instance 2、KVC造成的Crash 3、EXC_BAD_ACCESS 4、KVO引起的Crash 5、	
iOS防止崩溃机制以及底层的原理	
iOS防止崩溃机制以及底层的原理的有什么？ 吴杰 2011-11-19 在异常响中是否有这样的手机上，当刚刚刚下准备买货的一瞥的时候，突然到	
iOS捕获异常组件UncaughtExceptionHandler	
高级iOS开发工程师 iOS 捕获异常组件的函数是NSSetUncaughtExceptionHandler来行异常处理，但是无法处理内存访问错误，重复执行代	
iOS开发中常见崩溃原因汇总.你遇到了哪个？ONE_PERSON_ON...	
PROJECT -> Build Settings -> Architectures -> Valid Architectures 他的根本来源 armv7 armv7s (iOS6.0下) armv7s 去掉即 同样的操作 TARGETS -	
iOS常见crash问题及crash日志分析_inchad_的专栏	
第一、iOS常见Crash问题 1、插入元素 对于NSSetMutableArray、NSMutableDictionary、NSMutableSet等可变的对象，插入元素时Crash 正确做法是调用	
iOS常见崩溃和一些避免	
在日常开发中，当程序崩溃的时候是不是特别烦人，明明我的代码没有问题为何就崩溃了呢？ 又是因为什么原因导致的呢？该怎么避免？APP开发	
iOS常见崩溃情况以及bug调试解决方法	
本博客会不定期更新，我将在项目中常见的崩溃情况进行整理后分享给大家： 1.unrecognized selector，错误：这种情况很常见，给一个	
iOS Crash崩溃异常数据查看和分析_iJ.H.A的博文	
阅读过这篇文章的同学应该都知道，分析iOS异常问题的操作系统跟本机的仿真设备，以便方便调试和定位问题，你可以在崩溃异常报告的OSVersio	
iOS崩溃消息日志总结_justbestme的专栏	
iOS crash消息日志总结 当遇到应用程序崩溃以及因运行时系统生成对应的“crash”崩溃日志并存储在设备上，这份崩溃日志通常包含游戏内运行时相关	
iOS NSExcptionHandler异常处理	
iOS NSExcptionHandler异常处理 王耀辉的专栏 53	
前阵子，勇敢的小伙伴们大家好，这个是因为我在项目里集成了Bugly，但是不开不过问题，用户在使用过程中崩溃的时候能看见崩溃的原因，	
NSSetUncaughtExceptionHandler 处理异常	
在面试《高性能iOS应用开发》的时候，面试官提到了NSSetUncaughtExceptionHandler这个类，简单了解了一下，终于明白白灰灰，Bugly等软件，	
iOS开发 Crash总结与捕获_chenzhimin_blog	
(void*)newObjLoad ([super newObjLoad]); [self exceptionHandlerTest]; [self exceptionHandlerTest2]; [self exceptionHandlerTest3]; 异常处理流程1	
iOS崩溃日志分析(个人总结,最实用)学习路上	
iOS 崩溃日志分析(个人总结,最实用) 异常分析流程一般需要三个文件:crash日志,symbolicatecrash分析工具,iSYM符号表 在桌面创建一个crash文件夹.1	
iOS 报障 [NSSetUncaughtExceptionHandler initWithObjects]: attempt to insert nil object from objects[]	
错误提示: "" Terminating app due to uncaught exception 'NSSetInvalidArgumentException: reason: "" ([NSSetUncaughtExceptionHandler initWithObjects]:	
iOS崩溃 异常处理(NSExcptionHandler)	
iOS发布应用中对于异常捕捉捕获和处理 iOS开发中我们经常会遇到程序抛出异常造成的情况，如果是在测试的过程中，异常的情况一目了然，但是如果是	
iOS崩溃总结	
一、捕获iOS Crash 1、设置异常捕获点并运行 设置异常捕获点.png 编辑：设置Xcode异常捕获点并运行程序，发生Crash时，系统会自动弹出对话框，但仅	
iOS crash崩溃分析总结	
苹果官方文档: https://developer.apple.com/library/archivetchnologies/hot2004/t2123.html#/TSTRASHLOGEXCEPTION https://developer.apple.com/vibra	
iOS中iOS应用如何查看崩溃	
虽然大部分都来自于崩溃日志分析，但能崩溃消息也应当去面对面对的现实，既然崩溃消息无法回避，无法阻挡，那我们除了它也能避免发生呢，又Xcode	
iOS 崩溃日志收集及分析	
最近几天，项目中在增加推送功能，这时对推送推送SDK，微信大家也常用过，官方文档的集成步骤都很详细，集成也很简单，但是最近今天的项目有问	
iOS和安卓崩溃原因总结	
崩溃消息 内存管理问题 当iOS检测到内存访问错误，它的VM系统会发出堆内存警告消息，尝试回收一些内存 1在Debug模式下，可以主动将警告行执行...	
iOS崩溃异常处理(NSExcptionHandler)收集崩溃日志并分析	
1、在代码中添加一个收集崩溃日志代码 - (BOOL) application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions { NSS	
iOS异常处理及发送崩溃报告到服务器：NSSetUncaughtExceptionHandler	
在异常中常常会遇到崩溃消息，我们可以通以下两种途径来捕获崩溃消息信息发送到我们的服务器，这两种途径会捕获不同的崩溃消息：途径一： 读取 [NSSet	
通过下边的错误代码来执行： [NSSetUncaughtExceptionHandler: reason: "" ([NSSetUncaughtExceptionHandler initWithObjects]: attempt to insert nil obje	
好办法一揪取Crash，我们要在Crash产生时第一时间及时捕捉到它 作者：陶泽宇 2016-11-28 16:22 APP运行时Crash自动修复系统 前言 目前市面上关于A	
©2020 CSDN 皮肤主题: Age of AI 设计师: meimeielle 返回顶部	

关于我们 招贤纳士 广告服务 开发助手 400-600-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

公安备案号11010502030143 京ICP备19046568号 网文网 (2020) 1039-165号 经营网络违法违规信息举报中心 中国互联网违法和不良信息举报中心 不良信息举报中心

网络110报警服务 中国互联网举报中心 家长监护 Chrom商店下载 ©1999-2021 北京创新乐知网络技术有限公司 版权与免责声明 隐私权声明

京ICP备19046568号-22 京公网安备11010502030143

 hherima [关注](#)

 1
 0
 0
