

魏先宇的程序人生

多读，多写，多思，方有多得！

目录视图 摘要视图 RSS 订阅

个人资料



more2010wei

访问： 844813次
积分： 9460
等级： 5
排名： 第854名
原创： 122篇 转载： 115篇
译文： 10篇 评论： 92条

文章搜索

文章分类

- iphone开发之视图部分 (40)
- iphone开发之视图部分-scollview (3)
- iphone开发之数据部分 (38)
- iphone开发之内存部分 (19)
- iphone开发之框架部分 (16)
- iphone开发之功能实现部分 (18)
- iphone开发之多线程 (7)
- iphone开发之事件响应 (11)
- iphone开发之语法部分 (20)
- iphone开发之资料收集 (13)
- iphone开发之文件操作 (7)
- iphone开发之动画部分 (5)
- Quartz 2D (7)
- 关于调试 (8)
- 关于面向对象 (1)
- IOS6 (11)
- 数据库 (6)
- XCode (18)
- 资讯 (6)
- C和C++ (5)
- 生活的智慧 (1)

[下载CSDN移动客户端](#) [微信开发学习路线高级篇上线](#) [《Cocos3D与Shader从入门到精通》套餐限时优惠](#) [恭喜博主张安站新书上市](#)

解决EXC_BAD_ACCESS错误的一种方法--NSZombieEnabled

分类： [iphone开发之框架](#) 2012-05-14 20:30 31127人阅读 评论(2) 收藏 举报

[access](#) [xcode](#) [xcode4](#) [crash](#) [variables](#) [debugging](#)

我们做iOS 程序开发时经常用遇到 EXC_BAD_ACCESS 错误导致 Crash，出现这种错误时一般 Xcode 不会给我们太多的信息来定位错误来源，只是在应用 Delegate 上留下像Thread 1: Program received signal:"EXC_BAD_ACCESS"，让问题无从找起。

比如你对已释放的对象发送消息时就会出现， EXC_BAD_ACCESS，再如release 的对象再 release，release 那些autorelease 的对象等也会报这样的错。默认设置下 Xcode 不会给你定位具体是哪一行代码，不该去使用已释放的对象，或者release 用错了。

比如 UIViewController 子类中这样的代码：

```
[cpp] view plain copy print ?
01. static NSMutableArray*array;
02.
03. -(void)viewDidLoad
04. {
05.     [super viewDidLoad];
06.     array= [[NSMutableArray alloc] initWithCapacity:5];
07.     [array release]; //释放掉该数组
08. }
09.
10. - (void)viewWillAppear:(BOOL)animated{
11.     [array addObject:@"Hello"]; //使用释放掉的数组
12. }
```

上面的代码就会出现EXC_BAD_ACCESS 错误，但我执行时 Xcode 一出错却是定位在我在 AppDelegate 的 application:didFinishLaunchingWithOptions: 方法上的某行了，如果代码量多了，要查找具体问题非常难，但凭经验了。

不过NSZombieEnabled 环境变量可以帮我们的忙，就是当设置NSZombieEnabled环境变量后，一个对象销毁时会被转化为_NSZombie，设置NSZombieEnabled后，当你向一个已经释放的对象发送消息，这个对象就不会向之前那样Crash或者产生一个难以理解的行为，而是放出一个错误消息，然后以一种可预测的可以产生debug断点的方式消失，因此我们就可以找到具体或者大概是哪个对象被错误的释放了。

对 Xcode 设置了NSZombieEnabled 之后，Xcode 会明确定位在行[array addObject:@"Hello"]，然后控制台下报的错误信息是：

*** -[__NSArray addObject:]message sent to deallocated instance 0x6557370

如何设置 NSZombieEnabled 呢，在 Xcode3 和 Xcode4 下设置不一样，Xcode4 下设置很简单。

Xcode3 下 NSZombieEnabled 设置方法如下：

1. 在XCode左边那个Groups& Files栏中找到Executables，双击其中的一项，或者右键Get Info；
2. 切换到Arguments
3. 这里一共有两个框，在下面那个Variables to be set in theenvironment:点+号添加一项，Name里填NSZombieEnabled，Value填Yes，要保证前面的钩是选中的。

Xcode4 下设置 NSZombieEnabled 的方法：

你可以点击 Xcode4 菜单 Product -> Edit Scheme-> Arguments, 然后将点击”加号”，将 NSZombieEnabled 参数

证书 (2)
iOS5 (2)
越狱功能和私有API (4)
CoreText (2)
UIBezierPath (2)
NSCoding (2)
UIMenuController (1)
UIPageControl (1)
KVC (2)
KVO (2)
NSLock (1)
CALayer (2)
网络 (4)
抓包 (1)
ios开发规范 (2)
ios面试题集锦 (1)
block (12)
数据结构与算法 (4)
关键字 (1)
UIPasteboard (1)
设计模式 (1)
NSDate的使用 (1)
ARC (4)
Push (3)
CoreData (4)
Cocos2d-x (1)
JavaScript (1)
cocos2d (4)
cocos2d-HTML (4)
一步一步学Cocos2d-html5 (0)
storyboard (2)
Swift (0)
cocos2d-x 3.x游戏开发学习
笔记 (6)
pop开源库使用笔记 (1)
开源库使用笔记 (3)
性能优化 (3)
armv64 (1)
iphone6 (0)
iphone6的适配 (1)

文章存档

2015年03月 (7)
2014年12月 (2)
2014年11月 (3)
2014年10月 (2)
2014年09月 (3)

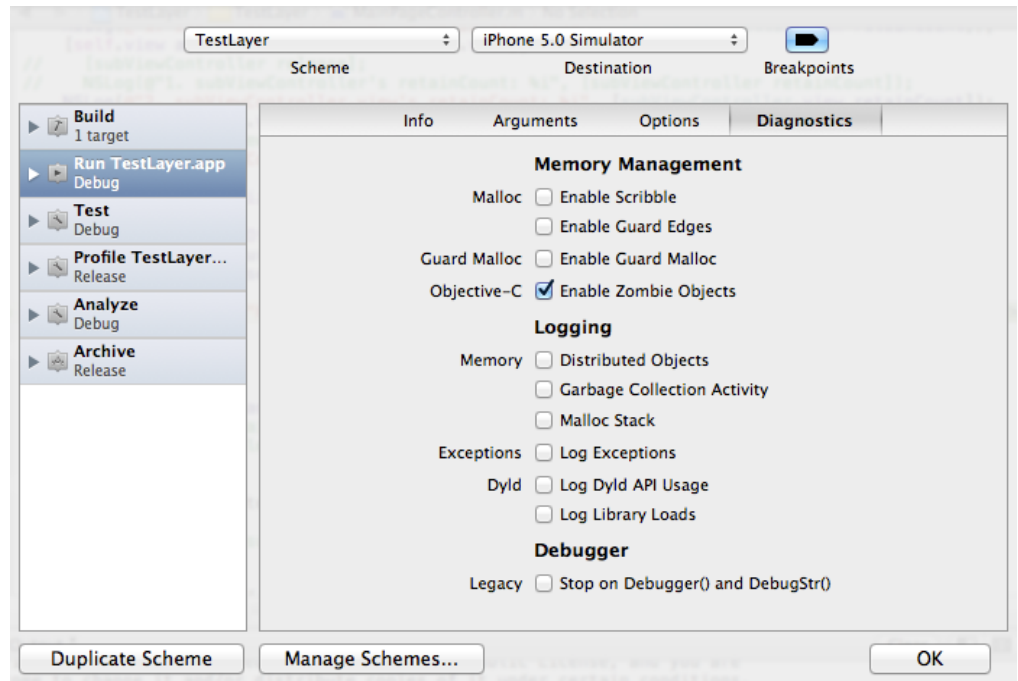
展开

阅读排行

ios的手势操作之UIGestu
(61126)
解决EXC_BAD_ACCESS
(31067)
关于userInteractionEnab
(21603)
NSString / NSMutableArray
(20837)
Xcode的控制台调试命令
(19656)
关于removeFromSuperv
(19108)
iOS 多线程 锁 互斥 同步
(17475)
IOS-应用之间调用
(15706)
关于iphone开发中的@pr
(15564)
IOS文件操作的两种方式

加到Environment Variables 窗口中, 后面的数值写上 "YES".

或者在 Xcode4 菜单 Product -> EditScheme -> Diagnostics 设置窗口中直接勾上Enable ZombieObjects 即可, Xcode 可用 cmd + shift + < 进到这个窗口。



Xcode4 已经考虑到了现在的要求, 所以提供了更便捷的设置的方式, 你也可以在这个窗口中设置其他一些参数, 你肯定能由此获得更多的帮助信息。

另外再说一下, 如果没有为 Xcode 设置 NSZombieEnabled, 像下面的代码或许可以正确执行, 打印出你所期望的结果"Hello"

```
[cpp] view plain copy print ?
01. static NSMutableArray*array;
02.
03. -(void)viewDidLoad
04. {
05.     [super viewDidLoad];
06.     array= [[NSMutableArray alloc]initWithCapacity:5];
07.     [array release];
08.     [array addObject:@"Hello"];//之所以不会crash, 是在于事件周期未完, 内存回收机制还没有执行, 没有
    真正的回收掉array的对象内存。
09.     NSLog(@"%@",[array objectAtIndex:0]);
10. }
```

但是一旦加上了NSZombieEnabled 设置, 上面的代码行 [array addObject:@"Hello"] 也将无法投机取巧了, 同样会得到错误提示:

*** -[__NSArrayM addObject:]:message sent to deallocated instance 0x6557370

即使该array 所指向的内存还是原来的数据也不能逃脱掉 NSZombieEnabled 的法眼。也就是之所以未设置 NSZombieEnabled 时上面代码能得到正确结果, 是因为, 虽然 [array release] 是标记为释放掉该内存块, 但是后面使用 array 时, 因为该指针指向的内存数据未被覆盖, 所以未出错, 这和C++ 的指针 delete 后的效果是一样的。

1. CocoaDev, 个人觉得讲Cocoa技术十分专业的网站之一, 下面的链接详细讲了讲NSZombieEnabled的原理。<http://www.cocoadev.com/index.pl?NSZombieEnabled>
2. 苹果官方的Mac OS X Debugging Magic,详细讲述了最为一个高级苹果程序员应该具备的调试技巧<http://developer.apple.com/library/mac/#technotes/tn2004/tn2124.html>
3. 其实还可以在Instruments中开启NSZombie选项, 这样就可以在Instruments中直接查看crash时候的callstack了: <http://www.markj.net/iphone-memory-debug-nszombie/>

最后提醒NSZombieEnabled只能在调试的时候使用, 千万不要忘记在产品发布的时候去掉, 因为NSZombieEnabled不会真正去释放dealloc对象的内存, 一直开启后果可想而知, 自重!

本文链接 http://unmi.cc/nszombieenabled-locate-exc_bad_access-error,来自 隔叶黄莺 Unmi Blog