

iOS开发 常见的崩溃闪退原因(12点)

前言

- 以下收集的崩溃原因，大家会觉得怎么也不会犯这错误，举个例子，我下面所写的 `NSDictionary *dic = @{@"key": nil};` 大部分可能认为他会崩溃，非常小心，如果是 `NSDictionary *dic = @{@"key": str}` 这种写法，大家有可能会忽略 `str` 是否为空，造成崩溃；
 - 把崩溃的原因写出来主要是想让大家记住这些，碰到这写的时候，能够小心，毕竟BUG都不是像我下面写的那么直观，都是会伪装的；
1. 字典的key或value为nil,字典的key只要为nil任何写法一定会崩，当value为nil的时候，只有 `NSDictionary *dic = @{@"key": nil}` 这种写法会崩，例如 `mutableDic[@"key"] = nil` 这种写法是不会崩的；

```
NSDictionary *dic = @{@"key": nil}; // 崩溃
```

```
[mutDic setObject:nil forKey:@"key"]; // 不崩溃  
mutDic[@"key"] = nil; // 不崩溃
```

2. 数组越界

```
NSArray *array= @[01, 02, 03];  
NSNumber *num = array[3]; // 崩溃
```

3. 内存溢出或泄露，分为下面几个方面

3.1 定时器没有释放或通知没有移除，或者循环引用，造成界面无法释放，内存增加

3.2 在for循环创建了很多局部变量，当遍历次数过多时造成内存急剧增加，崩溃，可以通过添加 `@autoreleasepool` 解决；

```
for (int i = 0; i < 5000000; i++) {  
    NSObject *obj = [[NSObject alloc] init]; // 内存暴增，局部变量没有释放
```

```

}

for (int i = 0; i < 5000000; i++) {
    @autoreleasepool {
        NSObject *obj = [[NSObject alloc] init]; // 内存减少, 运行时间差不多
    }
}

```

3.3 绘制图片，占用内存过大，例如

UIImageGetImageFromCurrentImageContext() 这些方法绘制大图片,或者加载大量高清晰体积大的图片

3.4 调用一些公司用C或C++语言写的静态库，由于不能对他们进行垃圾回收，造成内存泄露或溢出，用Instruments 来检查一下，如果你的项目是要支持长时间运行的，一定要仔细观察一下；

3.5 调用底层C语言框架，没对变量进行释放，造成内存泄露；

4. 遍历数组时，对数组进行操作

```

NSMutableArray *mutArr = @[@"a", @"b", @"c"].mutableCopy;
for (NSString *str in mutArr) {
    [mutArr addObject:@"d"]; // 崩溃
}

```

5. 截取字符串、数组、NSData越界

```

NSString *str = @"ABCDEFGH";
[str substringWithRange:NSMakeRange(0, 8)]; // 崩溃

```

6. 没有解决掉项目中出现的黄色警告和log警告提示，出现莫名其妙的崩溃，项目上线之前，也最好在release模式下跑一遍；

7. 类型错误，调用了没有实现的方法

```

NSDictionary *dic = @{@"key": @"a"};
NSArray *arr = dic[@"key"]; // arr其实是字符串, 但编译器不会报错
NSLog(@"%@", arr[0]); // 崩溃

```

8. 多个按钮同时点击造成崩溃，在AppDelegate中设置[UIButton appearance].exclusiveTouch = YES，避免按钮同时点击
9. iOS10以后，没有添加权限，访问相机，相册，联系人，麦克风等；
10. 使用最新的API,没有进行系统判别，在低版本的手机上崩溃；
11. 在swift中对空的类型强制解包，造成崩溃，一定要慎之又慎，尽量少用,不要为了方便，随随便便强制解包；
12. tableView没有注册cell,就调用 下面这方法

```
UITableViewCell *cell = [tableView  
dequeueReusableCellWithIdentifier:@"reuseIdentifier"  
forIndexPath:indexPath]` // 崩溃
```

结语

- 集成腾讯Bugly或者友盟来收集错误信息；
- 测试正常不代表APP Store 上的不崩溃，不怕麻烦，可以先用TestFlight测试一下，这样就能保证不出问题；