

# 关于野指针崩溃

## 1. 概述:

野指针应该在各个编程端都会存在的一个问题，那么，什么是野指针呢？

野指针：指针指向的对象已经被系统回收掉了，但是引用计数还未置0，也就是指针指向内存还在，这个指针就叫野指针，指向对象为僵尸对象，野指针指向了一个坏掉的内存区域。

对象被释放，说明该内存时刻可能由系统分配给其他对象使用。由于此时指针指向内存并不是原对象，所以尝试往该对象发送消息的话，就会造成程序崩溃

## 2. 野指针代码示例

(1)MRC下声明局部变量

```
Student *stu = [[Student alloc] init];
```

```
[stu setAge:10];
```

```
[stu release];
```

```
[stu setAge:10]; Thread 1: EXC_BREAKPOINT (code=
```

(2)ARC下对象属性修饰符为assign

```
@property (assign, nonatomic) Person *person;
```

```
self.person = 局部变量person1;
```

出了方法体后person1被系统回收

导致self.person指针变为野指针，这时候调用self.person的任何实例方法，都会造成程序崩溃

## 3. 总结

由于现在都是通过系统ARC进行内存管理，所以不考虑2(1)中情况，对于2(2)情况，项目中很容易发生错误的，就是assign用来修饰对象。其中，最明显的就是delegate了。

ARC出来之前，我们声明一个代理通常是用assign，ARC下鼓励代理的声明使用weak来替换assign，这个是有一定意义的。先说MRC下的delegate，通过assign修饰代理对象，当对象销毁后，系统并不会对其引用计数进行自减操作。也就是指针并不会被释放，指向内存已经坏死，这时候，如果代理事件触发，那么程序直接闪退，因为野指针指向的对象已经不存在具体对象。往其发送任何消息都会造成闪退。

而ARC下如果你用assign修饰代理，那么注意了，和MRC下代理一样，不需要使用该代理的时候，需要把delegate指针设为nil，让指针释放。一般，我们将这个操作写在dealloc方法里即可。

顺便说下，很容易出现这个问题的就是，使用了系统定时器NSTimer的代理方法了，页面销毁后系统管理不了定时器的内存，没办法释放定时器对象，如果用assign修饰定时器，那么页面关闭后，可能定时器就成野指针了，再次触发定时器代理的时候，也就是往僵尸对象delegate发送一条消息，程序直接崩溃。