

主页 > 开发频道 > 新手教学 > 基础知识

iOS中block的探究

发布于：2012-07-18 11:12 阅读数：14416

“Block是iOS4.0+ 和Mac OS X 10.6+ 引进的对C语言的扩展，用来实现匿名函数的特性。”

阅读器

iOS Objective-C block

/* ----- */

[5. Modify static local variable]

于是我们也可以推断出，静态局部变量是如何在block执行体中被修改的——通过指针。因为静态局部变量存在于数据段中，不存在栈展开后非法访存的风险。

上面中间代码片段与前一个片段的差别主要在于main函数里传递的是i的地址（&i），以及__main_block_impl_0结构体中成员i变成指针类型（int*）。然后在执行block时，通过指针修改值。当然，全局变量、静态全局变量都可以在block执行体内被修改。更准确地讲，block可以修改它被调用（这里是__main_block_func_0）时所处作用域内的变量。比如一个block作为成员变量时，它也可以访问同一个对象里的其它成员变量。

/* ----- */

[6. Implementation of __block variable]

那么，__block类型变量是如何支持修改的呢？

开发者通道

排行榜

代码库

图书库

网站库

发码区

工具库

招聘区

外包区

问答区

最近更新

1 iOS安全：黑客与反黑客
2012-12-20

2 使用Xcode和Instruments调试解决iC
2012-12-04

3 优秀开源代码解读：JS与iOS Native
2012-11-26

4 CocoaPods：一个Objective-C第三
2012-11-23

5 iOS开发——图片转PDF的实现方法
2012-11-22

6 iOS 6新特性UIActivityViewControll
2012-11-16

7 iOS 6中NSString新用法
2012-11-14

8 iOS中arc的设置与使用
2012-10-16

9 CGContext小记
2012-08-31

10 论坛用户mhmwadmiOS开发心得分
2012-08-16

推荐内容

热点内容

iOS中arc的设置与使用

CGContext小记

我们为int类型变量加上__block指示符，使得变量i可以在block函数体中被修改。

此时再看中间代码，会多出很多信息。首先是__block变量对应的结构体：

```
struct __Block_byref_i_0 {
    void *__isa;
    __Block_byref_i_0 *__forwarding;
    int __flags;
    int __size;
    int i;
};
```

由第一个成员__isa指针也可以知道__Block_byref_i_0也可以是NSObject。

第二个成员__forwarding指向自己，为什么要指向自己？指向自己是没有意义的，只能说有时候需要指向另一个__Block_byref_i_0结构。

最后一个成员是目标存储变量i。

此时，__main_block_impl_0结构如下：

```
struct __main_block_impl_0 {
    struct __block_impl impl;
    struct __main_block_desc_0* Desc;
    __Block_byref_i_0 *i; // by ref
    __main_block_impl_0(void *fp, struct __main_block_desc_0 *desc, __Block_byref_i_0 *i, int flags=0)
    : i(i->__forwarding) {
        impl.isa = &__NSConcreteStackBlock;
        impl.Flags = flags;
        impl.FuncPtr = fp;
        Desc = desc;
    }
};
```

__main_block_impl_0的成员变量i变成了__Block_byref_i_0 *类型。

对应的函数__main_block_func_0如下：

```
static void __main_block_func_0(struct __main_block_impl_0 *__cself) {
    __Block_byref_i_0 *i = __cself->i; // bound by ref
    (i->__forwarding->i) = 0; }
```

亮点是__Block_byref_i_0指针类型变量i，通过其成员变量__forwarding指针来操作另一个成员变量。:-)

而main函数如下：

```
int main()
{
    __block __Block_byref_i_0 i = {(void*)0, (__Block_byref_i_0 *)&i, 0, sizeof(__Block_byref_i_0), 1024};
    void (*blk)(void) = (void (*)(void))&__main_block_impl_0((void *)__main_block_desc_0_DATA, (struct __Block_byref_i_0 *)&i, 570425344);
    ((void (*)(struct __block_impl *))((struct __block_impl *)blk)->FuncPtr)((struct __block_impl *)blk);

    return 0;
}
```

通过这样看起来有点复杂的改变，我们可以修改变量i的值。但是问题同样存在：__Block_byref_i_0类型变量i仍然处于栈上，当block被回调执行时，变量i所在的栈已经被展开，怎么办？

在这种关键时刻，__main_block_desc_0站出来了：

```
static void __main_block_copy_0(struct __main_block_impl_0*dst, struct __main_block_impl_0*src) {__Block_object_assign((void*)&dst->i, (void*)&src->i, 8/*BLOCK_FIELD_IS_BYREF*/);}

static void __main_block_dispose_0(struct __main_block_impl_0*src) {__Block_object_dispose((void*)&src->i, 8/*BLOCK_FIELD_IS_BYREF*/);}

static struct __main_block_desc_0 {
    unsigned long reserved;
    unsigned long Block_size;
    void (*copy)(struct __main_block_impl_0*, struct __main_block_impl_0*);
    void (*dispose)(struct __main_block_impl_0*);
} __main_block_desc_0_DATA = { 0, sizeof(struct __main_block_impl_0), __main_block_copy_0, __main_block_dispose_0};
```

此时，__main_block_desc_0多了两个成员函数：copy和dispose，分别指向__main_block_copy_0和__main_block_dispose_0。

当block从栈上被copy到堆上时，会调用__main_block_copy_0将__block类型的成员变量i从栈上复制到堆上；而当block被释放时，相应地会调用__main_block_dispose_0来释放__block类型的成员变量i。

一会在栈上，一会在堆上，那如果栈上和堆上同时对该变量进行操作，怎么办？

这时候，__forwarding的作用就体现出来了：当一个__block变量从栈上被复制到堆上时，栈上的那个__Block_byref_i_0结构体中的__forwarding指针也会指向堆上的结构。

/* ----- */

本来还想继续写下去，结果发现文章有点长了。先到此。

原文链接：<http://blog.csdn.net/jasonblog/article/details/7756763>

Jason Lee @ Hangzhou

社区原帖：<http://www.cocoachina.com/bbs/read.php?tid=109750>



Objective-C 内存管理精髓



iOS新手入门视频教程



iOS中block的探究

猜你喜欢



关于block中使用
self的疑问



最近刚开始用
block 请问这个是



使用 block 的
UIAlertView -->



block
typeof(self)bself



MUBlockDelegate
——一个基于

- UIAlertView与UIActionSheet的Block实现方式
 - 由于xcode支持 block, 以后可以这样创建一个单
- block来加loading界面的问题
 - block 代替委托 传值 请教

现在评论

共1条评论

1 喜欢 社区



请输入你的评论

140

昵称 (必填)

发 布

按时间排序 | 新浪微博 | 腾讯微博



frank.feng (游客)

没怎么看懂。。。。

2012-7-19 10:19 顶 1

友言[?]



网站地图

关于我们

联系我们

合作云平台: 又拍云

京ICP备 11006519号 京ICP证 100954号

Copyright © 2008-2013 CocoaChina.com

资讯频道

开发相关
苹果相关

开发者频道

Mac开发
iPhone开发
新手教学
游戏开发
开发综合
用户体验
iPad开发

市场频道

AppStore研究
会员作品
软件销售
市场推广
上线经验
案例分析

下载频道

教学视频
电子文档
源码下载

开发者中心

应用排行
补充个人信息
华人应用大全

友情链接

麦芽地 雷锋网 工程师爸爸 安卓中文网 Nooidea.com | 装傻充愣 苹果发烧友 9RIA天地会 App汇 苹果发烧友 iPad网址导航
iPhone之家论坛 iPhoneSide 啃苹果论坛 LibFetion飞信 APLBBS苹果手机论坛 iPhoneTW 台湾iPhone俱乐 苹果fans
美图秀秀 iPhone 版 维以不永伤 远景苹果主题 麦课一班 MAC疯 爱苹果, 爱生活 MacDocks 我爱Podcast MacOrz macfav
苹果堂 爱Apps-每天一个好Ap 苹果贴士 苹果信息网 safari 5中文网