登录 | 注册

登录 | 注册

jeffasd的专栏

个人资料

jeffasd

访问： 132623次
积分： 5016
等级：
排名： 第3838名

原创： 239篇 转载： 859篇
译文： 18篇 评论： 21条

文章搜索

【公告】博客专栏旧貌换新颜 【hot】直播技术精选 主流编程语言图谱之二

NSPointerArray NSHashTable NSMapTable 详解

2016-01-12 17:16 146人阅读 评论(0) 收藏 举报

分类： iOS（643）

http://www.isaced.com/post-235.html

NSMapTable是早在Mac OS X 10.5（Leopard）的引入集合类。乍一看，这似乎是作为一个替换NSDictionary的存在，可以选择“strong”和“week”指针。在这篇文章中，我会告诉你除了为什么它也非常有用之外的还有垃圾回收机制以及它是如何做NSDictionary中不能（或不应该）做的事情。

Leopard 中更多的Cocoa API

可可增加了几个新的集合类在Mac OS X 10.5（Leopard）的。这些措施包括：

- NSPointerArray
- NSHashTable
- NSMapTable

NSPointerArray完全是新的，但大部分的 `NSHashTable` 和 `NSMapTable` 的功能之前可从 opaque Foundation C structs of the same names 看到。

在某些方面，这些新的类，像 `NSMutableArray` ， `NSMutableSet` 和的 `NSMutableDictionary` 一样工作，但是给了你使用“week”垃圾回收指针的选择。如果您使用的 Objective-C 2.0 垃圾回收机制，你应该知道什么是使用“week”指针，因此使用此选项的优势应该是清楚的。

NSPointerArray也可用于纯指针（指针不一定是Objective-C的类），但 `NSHashTable` 和的 `NSMutableArray` 类都需要它们的内容是Objective-C的对象。

iOS_下拉刷新 (5)
iOS_Git (4)
密码学 (2)
iOS_Wifi (10)
iOS_socket (8)
算法 (15)
数据结构 (3)
iOS_音视频 (14)
iOS_AutoLayout (4)
iOS_runtime (5)
OpenGL (37)

文章存档

2016年08月 (58)
2016年07月 (68)
2016年06月 (90)
2016年05月 (96)
2016年04月 (104)

展开

阅读排行

iOS8 PHAsset 照片框架 (1397)
iOS开发应用上架必读最 (1363)
iOS 获取图像的方式与坑 (1359)
CGBitmapContextCreate (1329)
iOS8 Layout Margins 详 (1323)
基于环信Demo3.0，实现 (1002)
git Updates were rejecte (983)
iOS 自定义 中间带突起圆 (940)
Xcode7基本操作 详解 (914)
iOS 下拉刷新 MJRefresh (911)

评论排行

利用AVPlayer播放iOS沙 (3)
iOS 视频旋转及平移详解 (2)
iOS info.plist 详解 (2)
iOS_直播类app_HTTP L (2)
修改navigationItem.back (1)
iOS开发RunTime之函数 (1)
iOS-Core-Animation之十 (1)
iOS 30多个iOS常用动画 (1)
iOS 获取图像的方式与坑 (1)
APNS推送服务证书制作 (1)

推荐文章

* 郭神带你真正理解沉浸式模式
* 优秀代码的格式准则
* Hadoop的数据仓库实践——OLAP与数据可视化（二）
* Android 视图篇——恼人的分割线留白解决之道
* 移动端开发者眼中的前端开发流程变迁与前后端分离

虽然在一般意义上，NSPointerArray and NSHashTable 被设计为可以替换 NSMutableArray and NSMutableSet 的角色（有序和无序阵列）。
NSMapTable则是不同的，因为它可以在你的设计中使用，而NSMutableDictionary不能（或不应该）。

NSPointerArray

NSPointerArray类是一个稀疏数组，工作起来与NSMutableArray相似，但可以存储NULL值，并且count方法会反应这些空点。可以用NSPointerFunctions对其进行各种设置，也有应对常见的使用场景的快捷构造函数strongObjectsPointerArray和weakObjectsPointerArray。

在能使用insertPointer:atIndex:之前，我们需要通过直接设置count属性来申请空间，否则会产生一个异常。另一种选择是使用addPointer:，这个方法可以自动根据需要增加数组的大小。

你可以通过allObjects将一个NSPointerArray转换成常规的NSArray。这时所有的NULL值会被去掉，只有真正存在的对象被加入到数组 — 因此数组的对象索引很有可能会跟指针数组的不同。注意:如果向指针数组中存入任何非对象的东西，试图执行allObjects都会造成EXC_BAD_ACCESS崩溃，因为它会一个一个的retain”对象”。

从调试的角度讲，NSPointerArray没有受到太多欢迎。description方法只是简单的返回了<NSConcretePointerArray: 0x17015ac50>。为了得到所有的对象需要执行[pointerArray allObjects]，当然，如果存在NULL的话会改变索引。

NSPointerArray性能特征

在性能方面，NSPointerArray真的非常非常慢，所以当你打算在一个很大的数据集合上使用它的时候一定要三思。在本测试中我们比较了使用NSNull作为空标记的NSMutableArray和使用了NSPointerFunctionsStrongMemory设置的NSPointerArray(这样对象会被适当的retain)。在一个有10,000个元素的数组中，我们每隔十个插入一个字符串”Entry %d”。此测试包括了用NSNull作为null填充的NSMutableArray。对于NSPointerArray，我们使用setCount:来代替
注意NSPointerArray需要的时间比NSMutableArray多了超过250x (!) 。这非常奇怪和意外。跟踪内存比较困难，似乎NSPointerArray更高效，但是因为我们使用同一个NSNull标记空对象，所以除了指针不该有更多的消耗。

NSDictionary的局限性

NSDictionary提供了key-to-object的映射。从本质上讲，NSDictionary中存储的object位置是由“key”来索引的。

由于对象存储在特定位置，NSDictionary中要求key的值不能改变（否则object的位
误）。为了保证这一点，NSDictionary中始终复制key到它私有位置。

这个key的复制行为也是NSDictionary如何工作的基础，但这也有一个限制：你可以只使用Objective-C对象作为NSDictionary的key，如果它支持NSCopying协议。此外，key应该是小且高效的，以至于复制的时候不会对CPU和内存造成负担。

这意味着，NSDictionary中真的只有适合“value”类型的对象作为key（如简短字符串和数字）。这不是离线的对象到对象的映射模型。

对象到对象的映射

最新评论

iOS info.plist 详解
jeffasd: 你把情况搞错了，读取的是nsbound 里面的数据，就是资源库的数据呀

iOS info.plist 详解
qq_35619794: 系统info.plist文件可写么可以用代码写入一些设置么

iOS 视频旋转及平移详解
jeffasd: 这个方法很垃圾 新的方法 可以用assetWrite 来实现

iOS 视频旋转及平移详解
zhf763120542: 你这个不通视频分辨率不同怎么处理，mainCompositionInst.renderSize=...

iOS_直播类app_HTTP Live Stre
qq_16564705: 自己定义的 加密和解密方式， 需要 给播放器 什么格式的 呢？ 244410894 求帮助

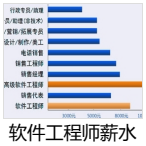
iOS_直播类app_HTTP Live Stre
qq_16564705: 写的 很好， 但是 有一点不太明白，就是如果我自己定义的 加密和 解密的 key 需要什么格式...

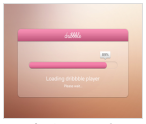
iOS block __block 关键字详解 O
探个究竟008: 哥们，你是从哪看到的Block的源码的啊？交流下，留个QQ吧，我的QQ：1358551681

OC 中深复制 和浅复制 详解
jeffasd: 这个说的挺详细的

iOS-Core-Animation之十二----性
jeffasd: 不错的文章

关于数组的几道重要的小算法
jeffasd: 不错得小算法


u盘制作

软件工程师薪水

计算机学校排名

自学ui

NSMapTable（顾名思义）更适合于一般意义的映射。这取决于它是如何构造的，NSMapTable可以处理的“key-to-object”样式映射的NSDictionary，但它也可以处理“object-to-object”的映射 - 也被称为“associative array”或简称为“map”。

例如，一个NSMapTable构造如下：

```
NSMapTable *keyToObjectMapping =
    [NSMapTable
        mapTableWithKeyOptions:NSMapTableCopyIn
        valueOptions:NSMapTableStrongMemory];
```

将会和NSMutableDictionary工作得一样一样的，复制其“key”，并retaining它的“object”。

一个纯粹的对象到对象(object-to-object)的映射可以构造如下：

```
NSMapTable *objectToObjectMapping =
    [NSMapTable mapTableWithStrongToStrongObjects];
```

一个对象到对象(object-to-object)的行为可能以前可以用NSDictionary来模拟，如果所有的key都是一个NSNumber包含于该映射的源对象的内存地址（不要笑，我见过这种情况），但这些内存地址都是奔波在外，Cocoa中首次提供了一个真正的对象到对象的映射NSMapTable。

NSMapTable的选项

NSMapTable提供的选项是由三部分组成：一个“memory option”(内存选项)，一个“personality option”和“copy in”标志。你可以为每个部分使用一个选项(如果没有提供一个选项的部分将会使用默认行为)，这个部分都是位标志(bit flag)(二进制 “or” 合并在一起)。

理论上，NSMapTable允许以下选项：

- NSMapTableStrongMemory (a "memory option")
- NSMapTableWeakMemory (a "memory option")
- NSMapTableObjectPointerPersonality (a "personality option")
- NSMapTableCopyIn (a "copy option")

NSMapTableStrongMemory是默认的“memory option”。然而，默认的“personality option”，默认“copy in”的行为没有名字那么这两个值可以被视为隐含在列表中。

memory option

Objective-C使用“strong”和“week”作为垃圾回收机制相关的术语，它可能不是很明显，这些选项可以在垃圾回收机制代码之外使用（苹果称它为手动内存管理）。

在垃圾回收机制外，他们被定义为：

- **strong**: 使用 retain 和 release
- **weak**: 不使用 retain 和 release

NSMapTable只允许NSPointerFunctionsOptions对应的Objective-C对象“personality option”。还有其他NSPointerFunctionsOptions “personality option”里的“strong”指针的行为不包括retain和release，但这些选项在NSMapTable都是不允许的。

关于使用垃圾回收机制的“week”之外的警告：
指针将不会被归零如在垃圾回收环境所以你必须要小心，不要取消引用指针，如果它被释放。

Personality options
该NSMapTableObjectPointerPersonality选项用来控制是否isEqualTo：和哈希对象中的方法添加的对象添加到集合时使用。

- NSMapTableObjectPointerPersonality指定
  对象的指针的值是用于直接比较和位移哈希生成（isEqualTo：和散列方法是不使用）。
- NSMapTableObjectPointerPersonality 不指定（默认行为）
  的哈希值与isEqualTo：方法会在调用的关键在确定的存储位置NSMapTable。这些方法的返回值不应改变（是不可变）为主要用在时间NSMapTable。

两行为暗示内容实现了 `NSObject` 的协议，所以在这个协议方法也可以在key和object调用。特别地，描述的方法可以在被调用NSMapTable包含密钥和对象无论使用的“Personality options”。该 `NSMapTable` 将只支持NSCoding如果所有的key和object实现了NSCoding协议了。

Copy options
如果NSMapTableCopyIn被指定，当 `NSCopying` 协议被加入时 `NSMapTable` 使用使自己的数据副本。如果不指定此选项（默认行为）将不会复制。

顶 0 踩 0

上一篇 ios中集合遍历方法的比较和技巧

下一篇 NSHashTable 和 NSMapTable的学习

我的同类文章

iOS（643）

- 某个服务的调用顺序 2016-09-19 阅读 5
- 状态 组合情况 太多的处理... 2016-09-09 阅读 14
- iOS URL encode 2016-09-07 阅读 23
- 使用友盟SDK提交Appstore... 2016-08-02 阅读 231
- RunLoop NSMachPort 详解 2016-07-25 阅读 69
- RunLoop个人学习 2016-07-25 阅读 12
- iOS description 函数一劳永... 2016-09-09 阅读 21
- Protobuffer和json深度对比 2016-09-08 阅读 25
- iOS制作framework以及引... 2016-08-30 阅读 61
- iOS 工具不错的网站 2016-07-27 阅读 21
- 深入理解RunLoop 2016-07-25 阅读 39

更多文章

猜你在找

疯狂IOS讲义之Objective-C面向对象设计
顾荣：开源大数据存储系统Alluxio（原Tachyon）的原
公开课：TCP协议的可靠性传输机制
Ceph—分布式存储系统的另一个选择
4.7. 存储类&作用域&生命周期&链接属性-C语言高级专是
疯狂IOS讲义之Objective-C面向对象设计
Docker与容器服务扩展机制 - 存储
顾荣：开源大数据存储系统Alluxio（原Tachyon）的原理
公开课：TCP协议的可靠性传输机制
Ceph—分布式存储系统的另一个选择
iOS7 漫谈基础集合类NSArray NSSet NSOrderedSet
NSHashTable的特性和使用
使用NSHashTable存储引用对象
NSHashTable的特性和使用
JS中NaNNULLundefined详解

中国无限制发行人民币
你的财富如何实现聚变？最后一次财富 分配机遇暗藏股市。中国3.0获利规则。
广告

查看评论

暂无评论

您还没有登录,请[登录]或[注册]

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack FTC
coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo
Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr
Angular Cloud Foundry Redis Scala Django Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持

京 ICP 证 09002463 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持

京 ICP 证 09002463 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved