



首页 &gt; iOS开发

# iOS开发多线程篇—NSOperation基本操作

2016-11-21 09:55 编辑: AllanHou 分类: iOS开发 来源: 文顶顶的博客

0 36

iOS开发 多线程

招聘信息: 移动应用开发项目经理

## 一、并发数

(1) 并发数:同时执?行的任务数.比如,同时开3个线程执行3个任务,并发数就是3

(2) 最大并发数: 同一时间最多只能执行的任务的个数。

(3) 最?大并发数的相关?方法

```
1 | - (NSInteger)maxConcurrentOperationCount;  
2 | - (void)setMaxConcurrentOperationCount:(NSInteger)cnt;
```

说明: 如果没有设置最大并发数, 那么并发的个数是由系统内存和CPU决定的, 可能内存多久开多一点, 内存少就开少一点。

注意: num的值并不代表线程的个数, 仅代表线程的ID。

提示: 最大并发数不要乱写 (5以内), 不要开太多, 一般以2~3为宜, 因为虽然任务是在子线程进行处理的, 但是cpu处理这些过多的子线程可能会影响UI, 让UI变卡。

## 二、队列的取消, 暂停和恢复

(1) 取消队列的所有操作

```
1 | - (void)cancelAllOperations;
```

提?:也可以调用NSOperation的- (void)cancel?法取消单个操作


(2) 暂停和恢复队列


```
1 | - (void)setSuspended:(BOOL)b; // YES代表暂停队列,NO代表恢复队列  
2 | - (BOOL)isSuspended; //当前状态
```

(3) 暂停和恢复的适用场合: 在tableview界面, 开线程下载远程的网络界面, 对UI会有影响, 使用户体验变差。那么这种情况, 就可以设置在用户操作UI (如滚动屏幕) 的时候, 暂停队列 (不是取消队列), 停止滚动的时候, 恢复队列。

## 三、操作优先级

### 热门资讯

 微信小程序的想象力与不可想象域  
点击量 8394

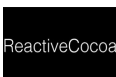
 39个优秀的Swift UI开源库 star平均数2527  
点击量 8184

 iOS 10和macOS中的卷积神经网络  
点击量 6143

 Java 之父求职被嫌年纪大, 硅谷公司现在喜  
点击量 5438

 移动端用户体验:底部导航  
点击量 5297

 UI设计中下拉刷新有什么讲究?  
点击量 4847

 ReactiveCocoa 5.0 初窥: 可能是最痛的一次  
点击量 2867

 iOS runtime实用篇--和常见崩溃say good-  
点击量 2639

 iOS开发网络—数据缓存  
点击量 2504

 应届生应聘iOS岗位的面试建议  
点击量 2166

### 综合评论

这个“他们不经常逛技术类的博客, 也不怎么看技术类的书籍, 只是习惯沉迷滴血雄鹰XF 评论了 你是码农还是优秀程序员? ...

marking。。。

onlyblue 评论了 iOS学习之路(资料推荐)...

查Storyboard资料的时候, 看到了这篇。。。依旧还是想学一下, 楼主观芒果街crj小屋 评论了 为什么iOS开发不需要Storyboard...

(1) 设置NSOperation在queue中的优先级,可以改变操作的执行优先级

```
1 - (NSOperationQueuePriority)queuePriority;
2 - (void)setQueuePriority:(NSOperationQueuePriority)p;
```

(2) 优先级的取值

```
1 NSOperationQueuePriorityVeryLow = -8L,
2 NSOperationQueuePriorityLow = -4L,
3 NSOperationQueuePriorityNormal = 0,
4 NSOperationQueuePriorityHigh = 4,
5 NSOperationQueuePriorityVeryHigh = 8
```

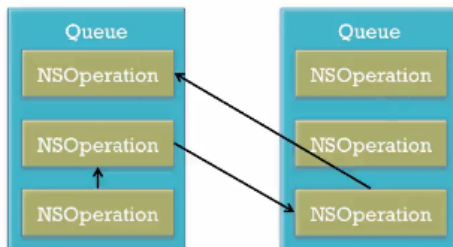
说明: 优先级高的任务, 调用的几率会更大。

## 四、操作依赖

(1) NSOperation之间可以设置依赖来保证执行顺序, 如一定要让操作A执行完后, 才能执行操作B, 可以像下面这么写

```
1 [operationB addDependency:operationA]; // 操作B依赖于操作A
```

(2) 可以在不同queue的NSOperation之间创建依赖关系



注意: 不能循环依赖 (不能A依赖于B, B又依赖于A)。

(3) 代码示例

```
#import "YYViewController.h"

@interface YYViewController ()

@end

@implementation YYViewController

- (void)viewDidLoad
{
    [super viewDidLoad];

    //创建NSInvocationOperation对象, 封装操作
    NSInvocationOperation *operation1=[[NSInvocationOperation alloc] initWithTarget:self selector:@selector(operation1)];
    NSInvocationOperation *operation2=[[NSInvocationOperation alloc] initWithTarget:self selector:@selector(operation2)];
    //创建对象, 封装操作
    NSBlockOperation *operation3=[NSBlockOperation blockOperationWithBlock:^(
        for (int i=0; i<5; i++) {
            NSLog(@"NSBlockOperation3--1----%@",[NSThread currentThread]);
        }
    )];
    [operation3 addExecutionBlock:^(
        for (int i=0; i<5; i++) {
            NSLog(@"NSBlockOperation3--2----%@",[NSThread currentThread]);
        }
    )];

    //设置操作依赖
    //先执行operation2, 再执行operation1, 最后执行operation3
    [operation3 addDependency:operation1];
    [operation1 addDependency:operation2];

    //不能是相互依赖
}
```

楼主 加个好友吧 问个问题

994291909 评论了 iOS蓝牙开发: 蓝牙连接和数据读写...

应届有人要?

MaydayiOS 评论了 应届生应聘iOS岗位的面试建议...

小白想问下, 这样子用同步队列, 在子线程中push一个新的ViewControllers的leeling 评论了 GCD实战练习: 使用串行队列实现简单的预加载...

让我们一起笑着活下去

MaydayiOS 评论了 ReactiveCocoa 5.0 初窥: 可能是最...

dota 现在还有人玩吗?

London1228 评论了 你是码农还是优秀程序员? ...

进入行业三年了, 也越来越觉得迷茫, 感觉再过5年后的自己的样子, 现在就不是开花的树 评论了 十年工龄的程序员为你揭示最危害程序员职业生涯的三大...

一直想要提高自己“估算时间”的能力, 但每次都急着先做了。。

心亦逸风 评论了 程序员的时间换算表...

## 相关帖子

分享某站长生活服务行业网站付费织梦模板(带手机端)

iOS HTTPS设置有没有必要做? 2016年马上过去了。

我的一个项目运行的时候报了一个alert。

macOS 这边就没人管管贴广告的么

《黑红方片草花机技巧》

《I37+28222+I4O》

《黑红方片草花机技术打法》

《I37+28222+I4O》

仿系统闹钟的功能的开发思路

App退出到后台后依然留在导航栏这种效果怎么做

《彩金翻牌机保单机仪器》

《I37+28222+I4O》 《彩金翻牌机保单接收器》

```
// [operation3 addDependency:operation1];
// [operation1 addDependency:operation3];

//创建NSOperationQueue
NSOperationQueue * queue=[[NSOperationQueue alloc]init];
//把操作添加到队列中
[queue addOperation:operation1];
[queue addOperation:operation2];
[queue addOperation:operation3];
}

-(void)test1
{
    for (int i=0; i<5; i++) {
        NSLog(@"NSInvocationOperation--test1--%@",[NSThread currentThread]);
    }
}

-(void)test2
{
    for (int i=0; i<5; i++) {
        NSLog(@"NSInvocationOperation--test2--%@",[NSThread currentThread]);
    }
}

@end
```

打印查看:

```
2014-06-25 23:21:42.812 03-NSOperation基本3[2123:1303] NSInvocationOperation--test2--<NSThread: 0x8b46ad0>{name = (null), num = 2}
2014-06-25 23:21:42.814 03-NSOperation基本3[2123:1303] NSInvocationOperation--test2--<NSThread: 0x8b46ad0>{name = (null), num = 2}
2014-06-25 23:21:42.815 03-NSOperation基本3[2123:1303] NSInvocationOperation--test2--<NSThread: 0x8b46ad0>{name = (null), num = 2}
2014-06-25 23:21:42.816 03-NSOperation基本3[2123:1303] NSInvocationOperation--test2--<NSThread: 0x8b46ad0>{name = (null), num = 2}
2014-06-25 23:21:42.817 03-NSOperation基本3[2123:1303] NSInvocationOperation--test2--<NSThread: 0x8b46ad0>{name = (null), num = 2}
2014-06-25 23:21:42.818 03-NSOperation基本3[2123:1303] NSInvocationOperation--test1--<NSThread: 0x8b46ad0>{name = (null), num = 2}
2014-06-25 23:21:42.819 03-NSOperation基本3[2123:1303] NSInvocationOperation--test1--<NSThread: 0x8b46ad0>{name = (null), num = 2}
2014-06-25 23:21:42.820 03-NSOperation基本3[2123:1303] NSInvocationOperation--test1--<NSThread: 0x8b46ad0>{name = (null), num = 2}
2014-06-25 23:21:42.821 03-NSOperation基本3[2123:1303] NSInvocationOperation--test1--<NSThread: 0x8b46ad0>{name = (null), num = 2}
2014-06-25 23:21:42.822 03-NSOperation基本3[2123:1303] NSInvocationOperation--test1--<NSThread: 0x8b46ad0>{name = (null), num = 2}
2014-06-25 23:21:42.881 03-NSOperation基本3[2123:1303] NSBlockOperation3-1--<NSThread: 0x8b46ad0>{name = (null), num = 2}
2014-06-25 23:21:42.881 03-NSOperation基本3[2123:1303] NSBlockOperation3-2--<NSThread: 0x8a3f680>{name = (null), num = 3}
2014-06-25 23:21:42.882 03-NSOperation基本3[2123:1303] NSBlockOperation3-2--<NSThread: 0x8a3f680>{name = (null), num = 3}
2014-06-25 23:21:42.882 03-NSOperation基本3[2123:1303] NSBlockOperation3-1--<NSThread: 0x8b46ad0>{name = (null), num = 2}
2014-06-25 23:21:42.883 03-NSOperation基本3[2123:1303] NSBlockOperation3-1--<NSThread: 0x8b46ad0>{name = (null), num = 2}
2014-06-25 23:21:42.883 03-NSOperation基本3[2123:1303] NSBlockOperation3-2--<NSThread: 0x8a3f680>{name = (null), num = 3}
2014-06-25 23:21:42.883 03-NSOperation基本3[2123:1303] NSBlockOperation3-2--<NSThread: 0x8a3f680>{name = (null), num = 3}
2014-06-25 23:21:42.883 03-NSOperation基本3[2123:1303] NSBlockOperation3-1--<NSThread: 0x8b46ad0>{name = (null), num = 2}
2014-06-25 23:21:42.900 03-NSOperation基本3[2123:1303] NSBlockOperation3-1--<NSThread: 0x8b46ad0>{name = (null), num = 2}
2014-06-25 23:21:42.884 03-NSOperation基本3[2123:1303] NSBlockOperation3-2--<NSThread: 0x8a3f680>{name = (null), num = 3}
```

A做完再做B, B做完才做C。

注意: 一定要在添加之前, 进行设置。

提示: 任务添加的顺序并不能够决定执行顺序, 执行的顺序取决于依赖。使用Operation的目的就是为了让开发人员不再关心线程。

## 五、操作的监听

可以监听一个操作的执行完毕

```
1 - (void (^)(void))completionBlock;
2 - (void)setCompletionBlock:(void (^)(void))block;
```

代码示例

第一种方式: 可以直接跟在任务后面编写需要完成的操作, 如这里在下载图片后, 紧跟着下载第二张图片。但是这种写法有的时候把两个不相关的操作写到了一个代码块中, 代码的可阅读性不强。

```
1 #import "YYViewController.h"
2
3 @interface YYViewController ()
4
5 @end
6
7 @implementation YYViewController
8
```

微博



CocoaChina

加关注

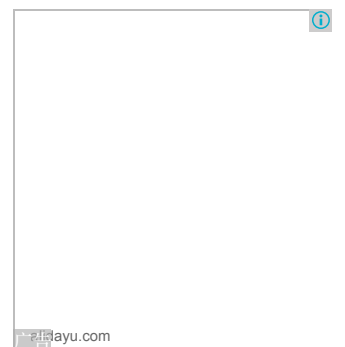
【如何在Xcode 8中更好地使用Storyboard】苹果在Xcode 8中为 Interface Builder 的界面做了非常伟大的改善, Size Classes 变得更加直观, Story Board 的使用也变得更加的便利, 还有一个完整度惊人的 Interface Builder 预览界面, 这对于那些对 interface Builder 的使用犹豫不决的人来说, 可能成为巨大的



24分钟前

转发(19) | 评论(2)

【应届生应聘iOS岗位的面试建议】每年都有大量的应届毕业生投入到iOS开发大军当中, 我过去3年也面试过



```

9 - (void)viewDidLoad
10 {
11     [super viewDidLoad];
12
13     //创建对象, 封装操作
14     NSBlockOperation *operation=[NSBlockOperation blockOperationWithBlock:^(
15         NSLog(@"-operation-下载图片-%@",[NSThread currentThread]);
16         //.....下载图片后继续进行的操作
17         NSLog(@"--接着下载第二张图片--");
18     )];
19
20     //创建队列
21     NSOperationQueue *queue=[[NSOperationQueue alloc]init];
22     //把任务添加到队列中 (自动执行, 自动开线程)
23     [queue addOperation:operation];
24 }
25
26 @end

```

第二种方式:

```

1 #import "YYViewController.h"
2
3 @interface YYViewController ()
4
5 @end
6
7 @implementation YYViewController
8
9 - (void)viewDidLoad
10 {
11     [super viewDidLoad];
12
13     //创建对象, 封装操作
14     NSBlockOperation *operation=[NSBlockOperation blockOperationWithBlock:^(
15         for (int i=0; i<10; i++) {
16             NSLog(@"-operation-下载图片-%@",[NSThread currentThread]);
17         }
18     )];
19
20     //监听操作的执行完毕
21     operation.completionBlock=^(
22         //.....下载图片后继续进行的操作
23         NSLog(@"--接着下载第二张图片--");
24     );
25
26     //创建队列
27     NSOperationQueue *queue=[[NSOperationQueue alloc]init];
28     //把任务添加到队列中 (自动执行, 自动开线程)
29     [queue addOperation:operation];
30 }
31
32 @end

```

打印查看:

```

2014-06-25 23:15:08.953 04-NSOperation基本5 (操作监听) [2060:1303] -operation-下载图片-<NSThread: 0x8d92de0>{name = (null), num = 2}
2014-06-25 23:15:08.955 04-NSOperation基本5 (操作监听) [2060:1303] -operation-下载图片-<NSThread: 0x8d92de0>{name = (null), num = 2}
2014-06-25 23:15:08.956 04-NSOperation基本5 (操作监听) [2060:1303] -operation-下载图片-<NSThread: 0x8d92de0>{name = (null), num = 2}
2014-06-25 23:15:08.957 04-NSOperation基本5 (操作监听) [2060:1303] -operation-下载图片-<NSThread: 0x8d92de0>{name = (null), num = 2}
2014-06-25 23:15:08.959 04-NSOperation基本5 (操作监听) [2060:1303] -operation-下载图片-<NSThread: 0x8d92de0>{name = (null), num = 2}
2014-06-25 23:15:08.961 04-NSOperation基本5 (操作监听) [2060:1303] -operation-下载图片-<NSThread: 0x8d92de0>{name = (null), num = 2}
2014-06-25 23:15:08.962 04-NSOperation基本5 (操作监听) [2060:1303] -operation-下载图片-<NSThread: 0x8d92de0>{name = (null), num = 2}
2014-06-25 23:15:08.963 04-NSOperation基本5 (操作监听) [2060:1303] -operation-下载图片-<NSThread: 0x8d92de0>{name = (null), num = 2}
2014-06-25 23:15:08.964 04-NSOperation基本5 (操作监听) [2060:1303] -operation-下载图片-<NSThread: 0x8d92de0>{name = (null), num = 2}
2014-06-25 23:15:08.965 04-NSOperation基本5 (操作监听) [2060:1303] -operation-下载图片-<NSThread: 0x8d92de0>{name = (null), num = 2}
2014-06-25 23:15:08.966 04-NSOperation基本5 (操作监听) [2060:1303] --接着下载第二张图片--

```

说明: 在上一个任务执行完后, 会执行operation.completionBlock=^{}代码段, 且是在当前线程执行 (2)。



微信扫一扫

订阅每日移动开发及APP推广热点资讯

公众号: CocoaChina

我要投稿

收藏文章

分享到:

0

上一篇: 如何在Xcode 8中更好地使用Storyboard

下一篇: IOS进阶之WKWebView

#### 相关资讯

来自一线开发者的Swift学习资源推荐

JSCore的基本使用

iOS开发UI篇—程序启动原理和UIApplication

使用苹果自带地图获取当前位置及周围信息

iOS 开发 -- 使用拦截器来取代基类

UICollectionView详解之自定义布局

iOS SDK 开发 -- 控制接口暴露

iOS 关于大小端以及一些数据补位

Objective-C copy, 看我就够了

不再安全的 OSSpinLock



广告

三网合一短信通道



我来说两句



你怎么看? 快来评论一下吧!

发表评论

所有评论 (0)

[关于我们](#) [商务合作](#) [联系我们](#) [合作伙伴](#)

北京融控科技有限公司版权所有

©2016 Chukong Technologies, Inc.

京ICP备 11006519号 京ICP证 100954号 京公网安备11010502020289



京网文[2012]0426-138号