

iOS-GCD多线程死锁案例

爱恨的潮汐 2019.06.10 16:08:32 字数 847 阅读 1,472

一、死锁场景:主线程调用主线程。

```
1 - (void)deadLockCase1 {
2     NSLog(@"1"); // 任务1
3     dispatch_sync(dispatch_get_main_queue(), ^{
4         NSLog(@"2"); // 任务2
5     });
6     NSLog(@"3"); // 任务3
7 }
8
9 控制台输出: 1, 后面就崩溃了。
```

原因: 从控制台输出可以看出, 任务2和任务3没有执行, 此时已经死锁了。因为dispatch_sync是同步的, 本身就会阻塞当前线程, 此刻阻塞了主线程, 而当前block又在等待主线程执行完毕, 从而形成了主线程等待主线程, 自己等自己的情况, 形成了死锁。

解决方法:

1、改用异步dispatch_async执行

```
1 NSLog(@"1"); // 任务1
2 dispatch_async(dispatch_get_main_queue(), ^{
3     NSLog(@"2"); // 任务2
4 });
5 NSLog(@"3"); // 任务3
6
7 控制台输出: 1 3 2
```

2、不在主线程中运行, 而是放在子线程中

```
1 NSLog(@"1"); // 任务1
2 dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_HIGH, 0), ^{
3     NSLog(@"2"); // 任务2
4 });
5 NSLog(@"3"); // 任务3
6
7 控制台输出: 1 2 3
```

注: 如果block中是刷新UI的操作, 则不能放在子线程中执行, 会crash

二、死锁场景2: (同步串行队列嵌套自己)

```
1 - (void)deadLockCase2 {
2     dispatch_queue_t aSerialDispatchQueue = dispatch_queue_create("com.test.deadlock.a");
3     NSLog(@"1"); //任务1
4     dispatch_sync(aSerialDispatchQueue, ^{
5         NSLog(@"2"); //任务2
6         dispatch_sync(aSerialDispatchQueue, ^{
7             NSLog(@"3"); //任务3
8         });
9         NSLog(@"4"); //任务4
10    });
11    NSLog(@"5"); //任务5
12 }
13 控制台输出: 1 2, 执行到2后面就崩溃了。
```

原因: 从控制台输出结果来看, 执行到任务2后, 就已经死锁了。因为该例子中两个GCD都是使用的同步方式, 而且还是同一个串行队列, 这就导致了和上一个例子一样, 自己在等待自己的情况, 形成了死锁。

解决方法:

1、将第二个GCD改为异步

```
1 - (void)deadLockCase2 {
2     dispatch_queue_t aSerialDispatchQueue = dispatch_queue_create("com.test.deadlock.a");
3     NSLog(@"1"); //任务1
4     dispatch_sync(aSerialDispatchQueue, ^{
5         NSLog(@"2"); //任务2
6         dispatch_async(aSerialDispatchQueue, ^{
7             NSLog(@"3"); //任务3
8         });
9         NSLog(@"4"); //任务4
10    });
11    NSLog(@"5"); //任务5
12 }
13 控制台输出: 1 2 4 5 3
```

然而, 将第一个GCD改为异步, 不能解决问题

```
1 - (void)deadLockCase2 {
2     dispatch_queue_t aSerialDispatchQueue = dispatch_queue_create("com.test.deadlock.a");
3     NSLog(@"1"); //任务1
4     dispatch_async(aSerialDispatchQueue, ^{
5         NSLog(@"2"); //任务2
6         dispatch_sync(aSerialDispatchQueue, ^{
7             NSLog(@"3"); //任务3
8         });
9         NSLog(@"4"); //任务4
10    });
11    NSLog(@"5"); //任务5
12 }
13 控制台输出:
14
15 5
16
17 2
```

虽然第一个GCD是异步的, 但是第二个GCD是同步的, 第二个GCD在等着第一个GCD结束, 而第一个GCD的block又在等着第一个GCD结束, 这样就形成了死锁。注: 对于以上将第二个GCD改为异步, 第一个GCD为同步的场景, 不会造成死锁, 是因为第二个GCD为异步, 它不用等待第一个GCD执行完毕, 它和第一个GCD是没有同步关系的。它是在第一个GCD执行的同时并发执行自己block的代码。

2、将两个GCD都改为异步

```
1 - (void)deadLockCase2 {
2     dispatch_queue_t aSerialDispatchQueue = dispatch_queue_create("com.test.deadlock.a");
3     NSLog(@"1"); //任务1
4     dispatch_async(aSerialDispatchQueue, ^{
5         NSLog(@"2"); //任务2
6         dispatch_async(aSerialDispatchQueue, ^{
7             NSLog(@"3"); //任务3
8         });
9         NSLog(@"4"); //任务4
10    });
11    NSLog(@"5"); //任务5
12 }
13 控制台输出:
14
15 1
16 5
17 2
18 4
19 3
```

3、使用不同的串行队列

```
1 - (void)deadLockCase2 {
2     dispatch_queue_t aSerialDispatchQueue1 = dispatch_queue_create("com.test.deadlock.a");
3     dispatch_queue_t aSerialDispatchQueue2 = dispatch_queue_create("com.test.deadlock.b");
4     NSLog(@"1"); //任务1
5     dispatch_sync(aSerialDispatchQueue1, ^{
6         NSLog(@"2"); //任务2
7         dispatch_sync(aSerialDispatchQueue2, ^{
8             NSLog(@"3"); //任务3
9         });
10        NSLog(@"4"); //任务4
11    });
12    NSLog(@"5"); //任务5
13 }
14 控制台输出:
15
16 1
17 2
18 3
19 4
20 5
```

三、死锁场景: 信号量阻塞主线程

```
1 - (void)viewDidLoad {
2     [super viewDidLoad];
3     dispatch_semaphore_t semaphore = dispatch_semaphore_create(0);
4     NSLog(@"semaphore create!");
5     dispatch_async(dispatch_get_main_queue(), ^{
6         dispatch_semaphore_signal(semaphore);
7         NSLog(@"semaphore plus 1");
8     });
9     dispatch_semaphore_wait(semaphore, DISPATCH_TIME_FOREVER);
10    NSLog(@"semaphore minus 1");
11 }
12 }
```

原因: 如果当前执行的线程是主线程, 以上代码就会出现死锁。因为dispatch_semaphore_wait(semaphore, DISPATCH_TIME_FOREVER)阻塞了当前线程, 而且等待时间是DISPATCH_TIME_FOREVER——永远等待, 这样它就永远的阻塞了当前线程——主线程。导致主线中的dispatch_semaphore_signal(semaphore)没有执行, 而dispatch_semaphore_wait一直在等待dispatch_semaphore_signal改变信号量, 这样就形成了死锁。

解决方法:

应该将信号量移到并行队列中, 如全局调度队列。以下场景, 移到串行队列也是可以的。但是串行队列还是有可能死锁的 (如果执行dispatch_semaphore_signal方法还是在对应串行队列中的话, 即之前提到的串行队列嵌套串行队列的场景)。

```
1 - (void)viewDidLoad {
2     [super viewDidLoad];
3     // Do any additional setup after loading the view, typically from a nib.
4     dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0), ^{
5         dispatch_semaphore_t semaphore = dispatch_semaphore_create(0);
6         NSLog(@"semaphore create!");
7         dispatch_async(dispatch_get_main_queue(), ^{
8             dispatch_semaphore_signal(semaphore);
9             NSLog(@"semaphore plus 1");
10        });
11        dispatch_semaphore_wait(semaphore, DISPATCH_TIME_FOREVER);
12        NSLog(@"semaphore minus 1");
13    });
14 }
15 上画从上到下顺序输出
```

以下代码运行结果如何?

```
1 - (void)viewDidLoad {
2     [super viewDidLoad];
3     NSLog(@"1");
4     dispatch_sync(dispatch_get_main_queue(), ^{
5         NSLog(@"2");
6     });
7     NSLog(@"3");
8 }
```

只输出: 1。后面就崩溃了 (主线程死锁, 因为viewDidLoad方法默认开了一条主线程, 然后又执行dispatch_sync(dispatch_get_main_queue(), ^{...});会导致你等我我等你, 结果导致死锁。

参考

四、下面代码打印顺序?

//同dispatch_queue_create函数生成的concurrent Dispatch Queue队列一起使用 dispatch_queue_t queue = dispatch_queue_create("test_queue", DISPATCH_QUEUE_CONCURRENT);

```
1 dispatch_async(queue, ^{
2     NSLog(@"----1-----%@", [NSThread.currentThread]);
3 });
4
5 dispatch_barrier_async(queue, ^{//这里模拟出屏障barrier函数等待
6     NSLog(@"----2----barrier-----%@", [NSThread.currentThread]);
7 });
8
9 dispatch_async(queue, ^{
10    NSLog(@"----3-----%@", [NSThread.currentThread]);
11 });
```

答案

1、2、3

6人点赞 > iOS自己写的面试题

更多精彩内容, 就在简书APP

微信扫一扫

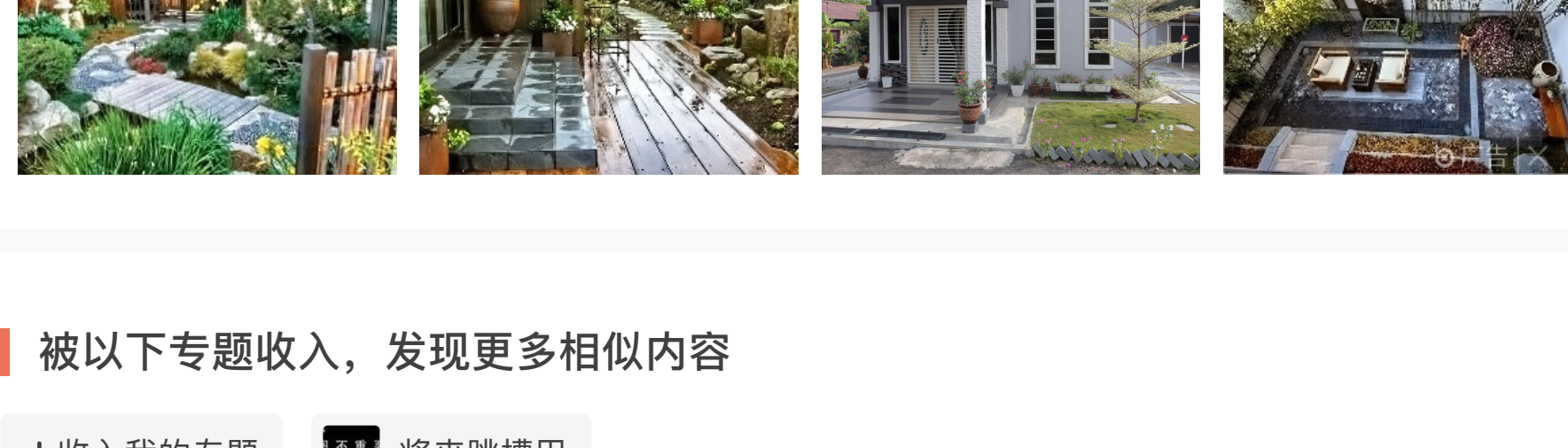
“小礼物走一走, 来简书关注我”

赞赏支持

还没有人赞赏, 支持一下

爱恨的潮汐

总资产300 共写了10.6W字 获得693个赞 共204个粉丝



被以下专题收入, 发现更多相似内容

+ 收入我的专题 将来能借用

推荐阅读

更多精彩内容 >

(转) iOS多线程:『GCD』详尽总结

本文转载自: 行走的少年郎的简书: iOS多线程:『GCD』详尽总结 本文用来介绍 iOS 多线程中 GCD 的相关知...

逍遥的游子 阅读 681 评论 0 赞 10

iOS多线程编程 基本知

iOS多线程编程 基本知识 1. 进程 (process) 进程是指在系统中正在运行的一个应用程序, 就是一段程序的执...

陈无山 阅读 2,230 评论 0 赞 9

工位出租600元/月

16年9月余姚丹山赤水、河姆渡游

算是短途游吧。2天。余姚的丹山赤水, 还算比较农家吧, 爬山真的好累, 土家菜口味不错哦。河姆渡真的差评, 啥也没有还那...

巴里问道 阅读 223 评论 0 赞 0

尾盘拉升暗示一大信号, 下周罕见走势即将上演!

周五, 沪深两市开盘涨跌不一, 随后市场呈现较为分化的走势, 沪指在银行、煤炭等权重板块的带领下继续低位震荡, 中小创则受...

牛眼看盘 阅读 51 评论 0 赞 1

推荐阅读

GCD的几种特殊用法

阅读 58

iOS 多线程 NSThread & GCD & NSOperation

阅读 273

面试官: 如何让主线程等待所有的子线程结束之后再执行? 我懵了

阅读 1,908

多线程基础知识+17个字判别死锁

阅读 188

iOS GCD之函数与队列

阅读 120

iOS开发自己的Framework之真相揭秘

程序员外包