

Kotlin 协程的suspend关键字到底有什么作用？进来看实验

原创

深海呐  已于 2022-03-15 18:24:38 修改  3363  收藏 6

分类专栏: [Android # Android进阶 # Kotlin](#) 文章标签: [kotlin 协程](#) [kotlin 协程](#) [kotlin suspend 协程](#) [suspend](#)

于 2022-03-15 17:53:51 首次发布

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/qq_39731011/article/details/123507414

版权



[Android 同时被 3 个专栏收录](#)

90 篇文章 3 订阅
订阅专栏



[Android进阶](#)

19 篇文章 5 订阅
订阅专栏



[Kotlin](#)

21 篇文章 1 订阅
订阅专栏

不久前收到一个粉丝的提问, 今天深海就来通过实验告诉大家 suspend 关键字到底有什么用。

首先这是测试代码:

build.gradle:

```
//协程  
implementation 'org.jetbrains.kotlinx:kotlinx-coroutines-core:1.1.1'
```

```

/**
 * author:xingHai.zhao(CSDN-深海呐)
 * duty: 协程举例
 */
@ExperimentalCoroutinesApi
fun main(args: Array<String>) {
    Test().test()
}

class Test {

    @ExperimentalCoroutinesApi
    fun test() {
        Thread {
            println("进入线程")
            GlobalScope.launch(Dispatchers.Unconfined) {
                println("协程1开始执行了!!!")
                timeConsuming()
            }
            println("到达线程中间")
            GlobalScope.launch(Dispatchers.Unconfined) {
                println("协程2开始执行了!!!")
            }
            while (true) {
                //模拟安卓主线程，防止线程死亡
            }
        }.start()
    }

    /**
     * author:xingHai.zhao
     * duty:模拟耗时方法
     */
    private suspend fun timeConsuming() {
        println("进入耗时任务,并遇到阻塞 -- -- -- ")
        delay(100)
        println("耗时任务开始执行 -- -- -- ")
    }
}

```

大概说一下这个代码做了什么：

开启一个线程，运行两个协程。

其中第一个协程调用了耗时方法 `timeConsuming()`，且该方法使用 `suspend` 修饰。

运行结果：

```
进入线程
协程1开始执行了!!!
进入耗时任务,并遇到阻塞 -- -- --
到达线程中间
协程2开始执行了!!!
耗时任务开始执行 -- -- --
CSDN @深海呐
```

怎么样？是不是有些诧异？

当协程1中的耗时任务遇到阻塞后，竟然直接去执行协程外的任务了。

并且协程2执行完之后又接着执行了协程1中后续耗时任务。

suspend功能总结：

那么集合上运行结果，suspend的功能显而易见。

当携程遇到阻塞方法时，如果该方法由suspend修饰，那么当前携程将进入挂起状态，

线程将跳过已挂起的携程，继续执行后面的任务，

当前面携程挂起状态结束后，线程将继续执行被挂起的线程后续的任务。（前提是这时候线程还存活）

为什么代码中有 while (true) 这个无限循环？

假如没这个循环的话，运行结果如下：

```
进入线程
协程1开始执行了!!!
进入耗时任务,并遇到阻塞 -- -- --
到达线程中间
协程2开始执行了!!!

Process finished with exit code 0
CSDN @深海呐
```

你会发现阻塞后的任务没有执行，因为线程死掉了。

这个 while (true) 可以保证线程不死，安卓主线程的 Handler原理亦是Loop的无限循环来保证主线程不死。

复制样例代码 运行出错？

如果你的报错是这个的话：

Command line is too long. In order to reduce its length classpath file can be used.
Would you like to enable classpath file mode for all run configurations of your project?



命令行过长。为了减少其长度，可以使用类路径文件。

是否要为项目的所有运行配置启用类路径文件模式？

CSDN @深海呐

这样解决：

解决：[Command line is too long. In order to reduce its length classpath file can be used._愿我如星君如月 ... 夜夜流光相皎洁 ...-CSDN博客](#)

好了，文章到这里就结束了，如果您有不同的见解或者有疑问的话欢迎到评论区留言。

如果您感觉深海写的不错的话，请给文章点个赞吧，感谢各位的支持！