

图文 074、无限制的调用方法是如何让线程的栈内存溢出的？

774 人次阅读 2019-09-12 07:00:00

详情 评论



狸猫技术

无限制的调用方法是如何让线程的栈内存溢出的？

狸猫技术专栏上新，基于真实订单系统的消息中间件（mq）实战，重磅推荐：

进店逛



相关频道



从 0 开
战高手
已更新1

未来3个月，我的好朋友原子弹大侠将带你一起，全程实战，360度死磕MQ

(点击下方蓝字进行试听)

从 0 开始带你成为消息中间件实战高手

重要说明：

如何提问：每篇文章都有评论区，大家可以尽情在评论区留言提问，我都会逐一答疑

(ps：评论区还精选了一些小伙伴对专栏每日思考题的作答，有的答案真的非常好！大家可以通过看别人的思路，启发一下自己，从而加深理解)

如何加群：购买了狸猫技术专栏的小伙伴都可以加入狸猫技术交流群。

(群里有不少一二线互联网大厂的助教，大家可以一起讨论交流各种技术)

具体加群方式请参见文末。

(注：以前通过其他专栏加过群的同学就不要重复加了)

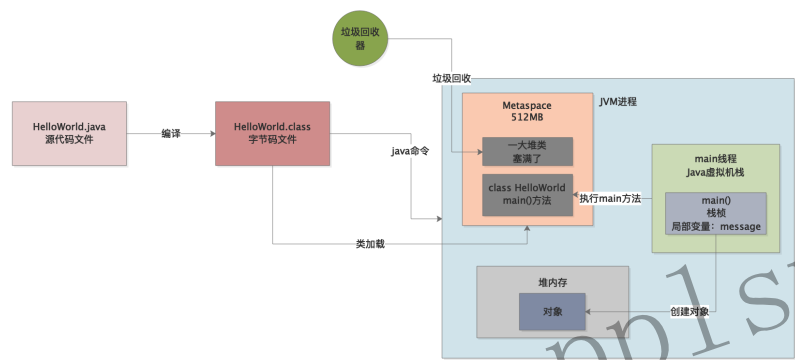
1、前文回顾

上一篇文章我们已经分析了Metaspace区域内存溢出的原理和两种情况，这篇文章我们就顺着JVM的运行原理继续分析一下，线程的栈内存是如何内存溢出的。

因为在JVM加载了我们写的类到内存里之后，下一步就是去通过线程执行方法，此时就会有方法的入栈出栈相关的操作，那么我们来分析一下线程的栈内存到底是因为什么原因会导致溢出呢？

2、一个线程调用多个方法的入栈和出栈

大家先回顾一下之前我们画好的图，那个图是一个相对较为完整的JVM运行原理图，如下所示。



现在我们来看下面的代码：

```
public class HelloWorld {
    public static void main(String[] args) {
        String message = "HelloWorld";
        System.out.println(message);
        sayHello("救火队队长");
    }
    public static void sayHello(String name) {
        System.out.println("你好，救火队队长");
    }
}
```

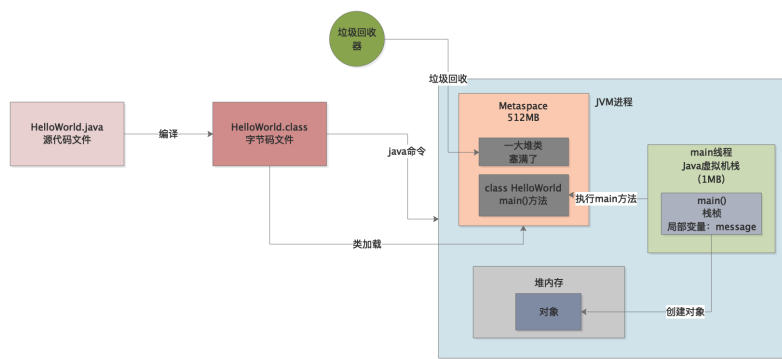
按照我们之前所说的，JVM启动之后，HelloWorld类被加载到了内存里来，然后就会通过main线程执行main()方法

此时在main线程的虚拟机栈里，就会压入main()方法对应的栈帧，里面就会放入main()方法中的局部变量。

大家看看上面的图，在图里是不是有main线程的虚拟机栈和main()方法的栈帧的概念？

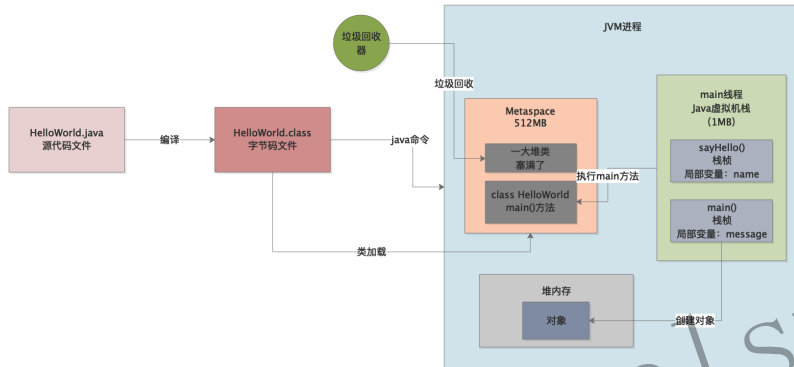
而且我们还知道一个概念，就是我们可以手动设置每个线程的虚拟机栈的内存大小的，一般来说现在默认都是给设置1MB

所以看下图，main线程的虚拟机栈内存大小一般也是固定的。

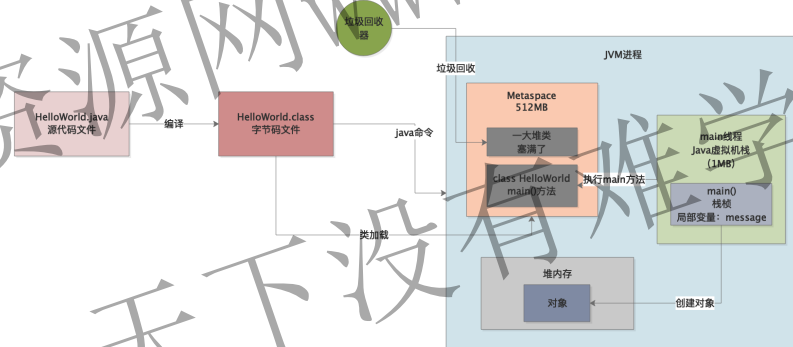


现在回过头思考一下上面的代码，代码中是不是在main()方法中又继续调用了sayHello()方法？

而且sayHello()方法中也会自己的局部变量，所以此时会继续将sayHello()方法的栈帧压入到main线程的虚拟机栈中去，如下图。



接着sayHello()方法如果运行完毕之后，就不需要为这个方法在内存中保存他的一些局部变量之类的东西了，此时就会将sayHello()方法对应的栈帧从main线程的虚拟机栈里出栈，如下图。



再接着，一旦main()方法自己本身也运行完毕，自然会将main()方法对应的栈帧也从main线程的虚拟机栈里出栈，这里我们就不在图里表示出来了。

3、一个重要的概念：每次方法调用的栈帧都是要占用内存的

在这里，要给大家明确一个重要的概念，那就是每个线程的虚拟机栈的大小是固定的，比如可能就是1MB，然后每次这个线程调用一个方法，都会将本次方法调用的栈帧压入虚拟机栈里，这个栈帧里是有方法的局部变量的。

虽然说一些变量和其他的一些数据占用不了太大的内存，但是大家要记得，每次方法调用的栈帧实际上也是会占用内存的！

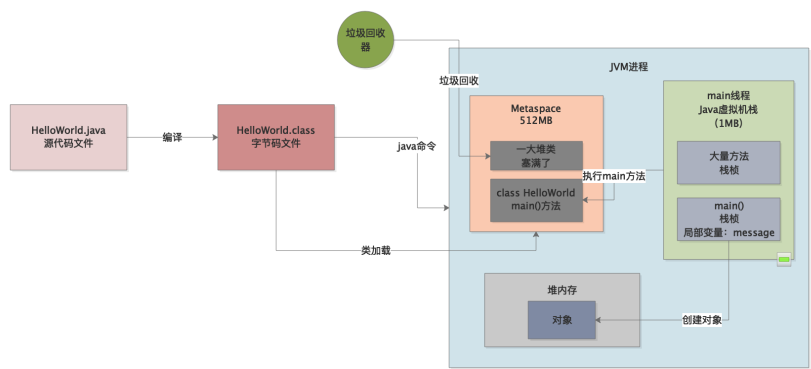
这是非常关键的一点，哪怕一个方法调用的栈帧就占用几百个字节的内存，那也是内存占用！

4、到底什么情况下会导致JVM中的栈内存溢出？

既然明确了上述前提之后，那么大家思考一下，到底什么情况下JVM中的栈内存会溢出呢？

其实非常简单，既然一个线程的虚拟机栈内存大小是有限的，比如1MB，那么假设你不停的让这个线程去调用各种方法，然后不停的把方法调用的栈帧压入栈中，是不是就会不断的占用这个线程1MB的栈内存？

如下图所示



那么如果不停的让线程调用方法，不停的往栈里放入栈帧，此时终有一个时刻，大量的栈帧会消耗完毕这个1MB的线程栈内存，最终就会导致出现栈内存溢出的情况。

5、一般什么情况下会发生栈内存溢出？

那么一般什么情况下会发生栈内存溢出呢？

通常而言，哪怕你的线程的虚拟机栈内存就128KB，或者256KB，通常都是足够进行一定深度的方法调用的。

但是如果说你要是走一个递归方法调用，那就不一定了，看下面的代码。

```
public static void sayHello(String name) {
    sayHello(name);
}
```

一旦出现上述代码，一个线程就会不停的调用同一个方法，即使是同一个方法，每一次方法调用也会产生一个栈帧压入栈里，比如说对sayHello()进行100次调用，那么就会有100个栈帧压入中。

所以如果疯狂的运行上述代码，就会不停的将sayHello()方法的栈帧压入栈里，最终一定会消耗掉线程的栈内存，引发内存溢出。

所以一般来说，其实引发栈内存溢出，往往都是代码里写了一些bug才会导致的，正常情况下发生的比较少。

6、今日文章总结

今天我们分析了栈内存溢出的根本原理和可能触发的一个场景，就是方法递归调用

但是一般来说，其实只要注意一下代码的编写，避免出现无限制的方法递归，就一般可以避免栈内存的溢出。

End

专栏版权归公众号狸猫技术窝所有

未经许可不得传播，如有侵权将追究法律责任

如何加群？

添加微信号：Lvqu0715_（微信名：绿小九），狸猫技术窝管理员

发送 Jvm专栏的购买截图

由于是人工操作，发送截图后请耐心等待被拉群