虚拟机字节码执行引擎

笔记本： jvm
创建时间： 2018/10/7/周日 13:23 **更新时间：** 2018/10/8/周一 19:10
作者： 1634896520@qq.com

虚拟机字节码执行引擎

执行引擎是Java虚拟机最核心的组成部分之一。“虚拟机”是一个相对于“物理机”的概念，这两种机器都有代码执行能力，其区别是物理机的执行引擎是直接建立在处理器、硬件、指令集和操作系统层面上的，而虚拟机的执行引擎则是由自己实现的，因此可以自行制定指令集与执行引擎的结构体系，并且能够执行那些不被硬件直接支持的指令集格式。

一、运行时栈帧结构

- 栈帧用于支持虚拟机进行方法调用和方法执行的数据结构，它是虚拟机运行时数据区中的虚拟机栈的栈元素。
- 栈帧存储了方法的局部变量表、操作数栈、动态连接和方法返回地址等信息。
- 每一个方法从调用开始到执行完成的过程，都对应着一个栈帧在虚拟机栈里面从入栈到出栈的过程。
- 对于执行引擎来说，在活动线程中，只有位于栈顶的栈帧才是有效的，称为当前栈帧，与这个栈帧相关联的方法称为当前方法。执行引擎运行的所有字节码指令都只针对当前栈帧进行操作。

（一）局部变量表

- 局部变量表示一组变量值存储空间，用于存放方法参数和方法内部定义的局部变量。
- 局部变量表的容量以变量槽为最小单位（Slot）
- 每个Slot都应该能存放一个boolean\byte\char\short\int\float\reference或returnAddress
- 虚拟机通过索引定位的方式使用局部变量表，索引值是从零开始到最大的Slot数量。

（二）操作数栈

- 操作数栈也称作“操作栈”，它是一个先进后出栈。
- 同局部变量表一样，操作栈的最大深度也在编译的时候写入Code属性的max_stacks的数据项中。
- 操作数栈的每一个元素都可以是任意的Java数据类型
- 在方法执行任何时候，操作数栈的深度都不会超过max_stacks数据项中设定的最大值。
- 操作数栈中的元素的数据类型，都应该与字节码指令的序列严格匹配。
- Java虚拟机的解释执行引擎称为“基于栈的执行引擎”，其中所指的“栈”就是操作数栈。

（三）动态连接

每个栈帧都包含一个指向运行时常量池中该栈帧所属方法的引用，持有这个引用是为了支持方法调用过程中的动态连接。

（四）方法返回地址

当一个方法开始执行后，只有两种方式可以退出这个方法：

- 正常完成出口：

执行引擎遇到任意一个方法返回的字节码指令，这时候可能会有返回值传递给上层的调用者，是否有返回值和是何种返回值类型都将根据遇到何种方法返回指令来决定

- 异常完成出口：

在方法执行过程中遇到了异常，并且在遇到这个异常时没有在方法体内得到处理，也许是Java虚拟机异常，还是代码异常，只要在本方法的异常表中没有搜到相应的异常处理，那么这种退出就叫异常完成出口。

（五）附加信息

虚拟机规范中允许具体的虚拟机添加一些规范里没有描述的信息到栈帧中。
在实际开发中，一般把动态连接、方法返回地址、其他附加信息全部归为一类，成为栈帧信息。

二、方法调用

①方法调用并不等于于方法执行，方法调用阶段唯一的任务就是确定被调用方法的版本，暂时还不涉及方法内部的具体运行过程。
②一切方法调用在class文件里面存储的都只是符号引用，而不是方法在实际运行时内存布局中的入口地址（相当于之前说的直接引用）。

更多详细参考。。。
方法解析和分派：
https://www.jianshu.com/p/100c44cca0d3
基于栈的字节码解释执行引擎：
https://www.jianshu.com/p/20c2b6f5fe59