

JVM加载Java类

1 加载步聚

2 java语言类型

3 加载

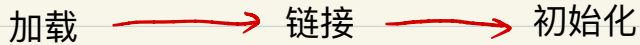
4 链接

5 初始化

6 总结

JVM如何加载Java类

1. 加载步骤:



2. java语言类型:

基本类型: byte short int long float double char boolean

引用类型: 类、接口、数组类、泛型参数

泛型参数: 编译过程中被擦除, 故不在JVM中

数组类是由 Java 虚拟机直接生成的, 类和接口则有对应的字节流。

无论是直接生成的数组类, 还是加载的类, 接口, Java 虚拟机都需要对其进行链接和初始化

3. 加载

加载, 是指查找字节流, 并且据此创建类的过程。对于数组类来说, 它并没有对应的字节流, 而是由 Java 虚拟机直接生成的。对于其他的类来说, Java 虚拟机则需要借助类加载器来完成查找字节流的过程。

3. 1 双亲委派模式

当加载java类的时候，根据双亲委派模式，每当一个类加载器接收到加载请求时，它会先将请求转发给父类加载器。在父类加载器没有找到所请求的类的情况下，该类加载器才会尝试去加载。

最高级的加载器为启动类加载器，除了启动类加载器之外，其他的类加载器都是 `java.lang.ClassLoader` 的子类，除了启动类加载器之外，另外两个重要的类加载器是扩展类加载器（`extension class loader`）和应用类加载器（`application class loader`），均由 Java 核心类库提供。

3. 2 平台类加载器

Java 9 引入了模块系统，并且略微更改了上述的类加载器。扩展类加载器被改名为平台类加载器（`platform class loader`）。Java SE 中除了少数几个关键模块，比如说 `java.base` 是由启动类加载器加载之外，其他的模块均由平台类加载器所加载。

3. 3 类的唯一性

类的唯一性是由类加载器实例以及类的全名一同确定的。即便是同一串字节流，经由不同的类加载器加载，也会得到两个不同的类。在大型应用中，我们往往借助这一特性，来运行同一个类的不同版本

4. 链接

链接，是指将创建成的类合并至 Java 虚拟机中，使之能够执行的过程。分为三个阶段

三阶段：

验证：确保被加载类能够满足 Java 虚拟机的约束条件

准备：为被加载类的静态字段分配内存

解析：将编译器生成的包含目标方法所在类的名字、目标方法的名字、接收参数类型以及返回值类型的符号引用解析为实际引用。

5. 初始化

为标记为常量值的字段赋值，以及执行 `< clinit >` 方法的过程。Java 虚拟机会通过加锁来确保类的 `< clinit >` 方法仅被执行一次。

初始化触发情况：

当虚拟机启动时，初始化用户指定的主类；

当遇到用以新建目标类实例的 `new` 指令时，初始化 `new` 指令的目标类；

当遇到调用静态方法的指令时，初始化该静态方法所在的类；当遇到访问静态字段的指令时，初始化该静态字段所在的类；

子类的初始化会触发父类的初始化；

如果一个接口定义了 `default` 方法，那么直接实现或者间接实现该接口的类的初始化，会触发该接口的初始化；

使用反射 API 对某个类进行反射调用时，初始化这个类；

当初次调用 `MethodHandle` 实例时，初始化该 `MethodHandle` 指向的方法所在的类

6总结

加载是指查找字节流，并且据此创建类的过程。加载需要借助类加载器，在 Java 虚拟机中，类加载器使用了双亲委派模型，即接收到加载请求时，会先将请求转发给父类加载器。

链接，是指将创建成的类合并至 Java 虚拟机中，使之能够执行的过程。链接还分验证、准备和解析三个阶段。其中，解析阶段为非必须的。

初始化，则是为标记为常量值的字段赋值，以及执行 `< clinit >` 方法的过程。类的初始化仅会被执行一次，这个特性被用来实现单例的延迟初始化。