

初识java代码的运行

- 1 Java代码的运行方式
- 2 JDK和JRE区别
- 3 Java为何需要在虚拟机运行
- 4 用Java 虚拟机的好处
- 5 JAVA具体如何运行java字节码
- 6 两种编译方式比较
- 7 实验(Java虚拟机将boolean看作 `int = 1`)

ACHEN

初识java代码的运行

1 Java代码的运行方式

Java 代码有很多种不同的运行方式：比如说可以在开发工具中运行，可以双击执行 jar 文件运行，也可以在命令行中运行，甚至可以在网页中运行。

java代码的运行离不开它自身的运行环境也就是JRE

2 JDK和JRE区别

JDK是Java的开发工具，它不仅提供了Java程序运行所需的JRE，还提供了一系列的编译，运行等工具，如javac, java, javaw等。

JRE只是Java程序的运行环境，它最核心的内容就是JVM（Java虚拟机）及核心类库。



ACHEN

1、java.exe:

运行java程序

2、javac.exe:

编译的，生成.class文件

3、javaw.exe:

跟java命令相对的，运行java命令时，会出现并保持一个console窗口，

程序中的信息可以通过System.out在console内输出，而运行javaw，

开始时会出现console，当主程序调用之后，console就会消失；

javaw 大多用来运行GUI程序

ACHEN

3 Java为何需要在虚拟机运行

Java 作为一门高级程序语言，它的语法非常复杂，抽象程度也很高。因此，直接在硬件上运行这种复杂的程序并不现实。所以呢，在运行 Java 程序之前，我们需要对其进行一番转换，使用虚拟机。

转换思路：

设计一个面向 Java 语言特性的虚拟机，并通过编译器将 Java 程序转换成该虚拟机所能识别的指令序列，也称 Java 字节码。之所以叫字节码，是因为 Java 字节码指令的操作码（opcode）被固定为一个字节。

```
# 最左列是偏移；中间列是给虚拟机读的机器码；最右列是给人读的
0x00: b2 00 02      getstatic java.lang.System.out
0x03: 12 03         ldc "Hello, World!"
0x05: b6 00 04      invokevirtual java.io.PrintStream.println
0x08: b1            return
```

4 用Java 虚拟机的好处

Java 虚拟机可以由硬件实现，但更多的是软件实现，这样编译为字节码的Java可以在不同平台上的JVM运行，一次编写，到处运行。这是第一个好处。

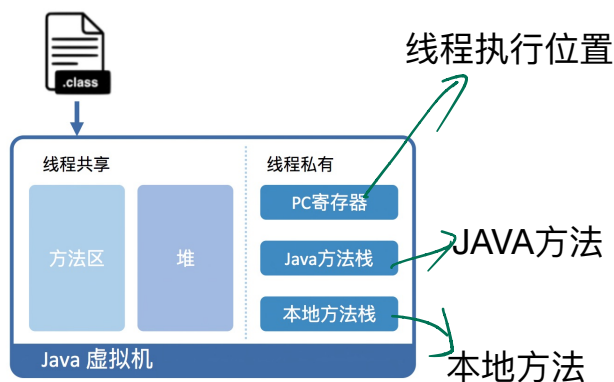
虚拟机的另外一个好处是它带来了一个托管环境（Managed Runtime）。这个托管环境能够代替我们处理一些代码中冗长而且容易出错的部分。其中最广为人知的当属自动内存管理与垃圾回收。

ACHEN

5 JAVA具体如何运行java字节码

虚拟机视角：

执行 Java 代码首先需要将它编译而成的 class 文件加载到 Java 虚拟机中。加载后的 Java 类会被存放于方法区 (Method Area)中，运行时，虚拟机执行方法区中代码



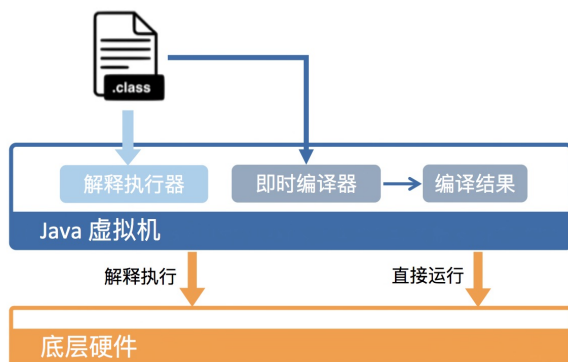
JAVA字节码的执行：

Java 字节码无法直接执行。因此，Java 虚拟机需要将字节码翻译成机器码。如何翻译：

在 HotSpot 里面，上述翻译过程有两种形式：第一种是解释执行，即逐条将字节码翻译成机器码并执行；第二种是即时编译（Just-In-Time compilation, JIT），即将一个方法中包含的所有字节码编译成机器码后再执行

ACHEN

6 两种编译方式比较



解释执行：无需等待编译

即时编译：实际运行速度更快

HotSpot 默认采用混合模式，综合了解释执行和即时编译两者的优点。它会先解释执行字节码，而后将其中反复执行的热点代码，以方法为单位进行即时编译。

HotSpot 内置了多个即时编译器：C1、C2 和 Graal

C1 又叫做 Client 编译器，面向的是对启动性能有要求的客户端 GUI 程序，采用的优化手段相对简单，因此编译时间较短。C2 又叫做 Server 编译器，面向的是对峰值性能有要求的服务器端程序，采用的优化手段相对复杂，因此编译时间较长，但同时生成代码的执行效率较高。

热点方法首先会被 C1 编译，而后热点方法中的热点会进一步被 C2 编译。

ACHEN

Java 语言和 Java 虚拟机看待 boolean 类型的方式不同，Java 虚拟机将boolean看作 `int = 1`;

实验:

1. 下载asmtools.jar

命令行:

```
hg clone http://hg.openjdk.java.net/code-tools/asmtools
cd asmtools/build
ant
```

asmtools-7.0-build/dist/asmtools-7.0.zip中得到
zip文件，解压zip

2. 将解压后的，asmtools.jar文件放到一个新建的Foo文件夹中

3. cd到Foo文件夹

4. 生成java文件:

```
echo 'public class Foo { public static void main(String[] args)
{ boolean flag = true; if (flag) System.out.println("Hello, Java!"); if
(flag == true) System.out.println("Hello, JVM!"); }}' > Foo.java
```

```
public class Foo {
    public static void main(String[] args) {
        boolean flag = true;
        if (flag) System.out.println("Hello, Java!");
        if (flag == true) System.out.println("Hello, JVM!");
    }
}
```

ACHEN

5. 编译为class文件
javac Foo.java

6.打开Foo.class文件

```
import java.io.PrintStream;

public class Foo
{
    public static void main(String[] paramArrayOfString)
    {
        int i = 1;
        if (i != 0) {
            System.out.println("Hello, Java!");
        }
        if (i == 1) {
            System.out.println("Hello, JVM!");
        }
    }
}
```

可以看出在虚拟机中，boolean看作int 为1

7.java Foo 运行

```
(base) [yunchenouyang@localhost Foo]$java Foo
Hello, Java!
Hello, JVM!
```

8.修改int为2

```
awk 'NR==1,/iconst_1/{sub(/iconst_1/, "iconst_2")}' 1'
FooAnother.jasm > Foo.jasm
```

9.编译

```
java -jar asmtools.jar jasm Foo.jasm
```

10.结果:

ACHEN


```
(base) [yunchenouyang@localhost Foo]$java Foo
Hello, Java!
```

由此映证了在java虚拟机中将boolean类型的true看为1

```
bash
Last login: Wed Feb 26 16:30:07 on ttys000
(base) [yunchenouyang@localhost ~]$cd /Users/yunchenouyang/Desktop/Code/Foo
(base) [yunchenouyang@localhost Foo]$
(base) [yunchenouyang@localhost Foo]$
(base) [yunchenouyang@localhost Foo]$echo 'public class Foo { public static void main(String[] args) {
boolean flag = true; if (flag) System.out.println("Hello, Java!"); if (flag == true) System.out.print
ln("Hello, JVM!"); }}' > Foo.java
(base) [yunchenouyang@localhost Foo]$
(base) [yunchenouyang@localhost Foo]$
(base) [yunchenouyang@localhost Foo]$javac Foo.java
(base) [yunchenouyang@localhost Foo]$
(base) [yunchenouyang@localhost Foo]$
(base) [yunchenouyang@localhost Foo]$java Foo
Hello, Java!
Hello, JVM!
(base) [yunchenouyang@localhost Foo]$java -jar asmtools.jar jdis Foo.class > FooAnother.jasm
(base) [yunchenouyang@localhost Foo]$$ awk 'NR==1,/iconst_1/{sub(/iconst_1/, "iconst_2")} 1' Foo.jasm.
1 > Foo.jasm
-bash: $: command not found
(base) [yunchenouyang@localhost Foo]$awk 'NR==1,/iconst_1/{sub(/iconst_1/, "iconst_2")} 1' Foo.jasm.1
> Foo.jasm
awk: can't open file Foo.jasm.1
source line number 1
(base) [yunchenouyang@localhost Foo]$awk 'NR==1,/iconst_1/{sub(/iconst_1/, "iconst_2")} 1' FooAnother.
jasm > Foo.jasm
(base) [yunchenouyang@localhost Foo]$
(base) [yunchenouyang@localhost Foo]$
(base) [yunchenouyang@localhost Foo]$java -jar asmtools.jar jasm Foo.jasm
(base) [yunchenouyang@localhost Foo]$
(base) [yunchenouyang@localhost Foo]$
(base) [yunchenouyang@localhost Foo]$java Foo
Hello, Java!
(base) [yunchenouyang@localhost Foo]$
```

ACHEN