

在笔试中，经常会考到 ++i 与 i++ 相关的陷阱题，这篇文章讲从字节码的角度，彻底帮你弄懂背后发生的事情。

0x01 看一道笔试题

```
public static void foo() {  
    int i = 0;  
    for (int j = 0; j < 50; j++)  
        i = i++;  
    System.out.println(i);  
}
```

输出结果是 0，而不是 50

关于 i++ 和 ++i 的区别，稍微有经验的程序员都或多或少都是了解的。听过了很多道理，依旧过不好这一生，我们从字节码的角度来彻底分析一下

```

public static void foo();
    0: iconst_0
    1: istore_0
    2: iconst_0

    3: istore_1
    4: iload_1
    5: bipush      50
    7: if_icmpge    21

    10: iload_0
    11: iinc         0, 1
    14: istore_0

    15: iinc         1, 1
    18: goto         4

    21: getstatic    #3          //
Field java/lang/System.out:Ljava/io/PrintStream;
    24: iload_0
    25: invokevirtual #4          //
Method java/io/PrintStream.println:(I)V
    28: return

```

对应 `i = i++;` 的字节码是 10 ~ 14 行：

- 10: `iload_0` 把局部变量表 `slot = 0` 的变量(`i`)加载到操作数栈上
- 11: `iinc 0, 1` 对局部变量表 `slot = 0` 的变量(`i`)直接加 1，但是这时候栈顶的元素没有变化，还是 0
- 14: `istore_0` 将栈顶元素出栈赋值给局部变量表 `slot = 0` 的变量，也就是 `i`。在这时，局部变量 `i` 又被赋值为 0 了，前面的 `iinc` 指令对 `i` 的加一操作前功尽弃。

如果要用伪代码来理解 $i = i++$ ，应该是下面这样的

```
tmp = i;  
i = i + 1;  
i = tmp;
```

0x02 ++i 又会怎么样

把代码稍作修改，如下

```
public static void foo() {  
    int i = 0;  
    for (int j = 0; j < 50; j++)  
        i = ++i;  
    System.out.println(i);  
}
```

来看对应的字节码

```

public static void foo();
    0: iconst_0
    1: istore_0
    2: iconst_0
    3: istore_1
    4: iload_1
    5: bipush          50
    7: if_icmpge        21

    10: iinc             0, 1
    13: iload_0
    14: istore_0

    15: iinc             1, 1
    18: goto             4
    21: getstatic        #3                      //
Field java/lang/System.out:Ljava/io/PrintStream;
    24: iload_0
    25: invokevirtual   #4                      //
Method java/io/PrintStream.println:(I)V
    28: return

```

可以看到 `i = ++i;` 对应的字节码还是 10 ~ 14 行，与 `i++` 的字节码对比如下图：

可以看出 `i = ++i;` 先对局部变量表 `slot = 0` 的变量加 1，然后才把它加载到操作数栈上，随后又从操作数栈上出栈赋值给了局部变量表，最后写回去的值也是最新的值。

画出整个过程的局部变量表和操作数栈的变化如下：

如果要用伪代码来理解 `i = ++i`，应该是下面这样的

```
i = i + 1;
tmp = i;
i = tmp;
```

0x03 看一道难一点的题目

```
public static void bar() {
    int i = 0;
    i = i++ + ++i;
    System.out.println("i=" + i);
}
```

输出是什么？

同样我们以字节码的角度来分析，add 指令的第一个参数值为 0，第二个参数值为 2，最终输出的结果为 2，详细的分析过程我画了一个简单的过程图，如下：

```
public static void bar();
    0: iconst_0
    1: istore_0

    2: iload_0
    3: iinc          0, 1
    6: iinc          0, 1
    9: iload_0
   10: iadd
   11: istore_0
```

用伪代码的方式就是，会不会更好理解一些？

```
i = 0;

tmp1 = i;
i = i + 1;

i = i + 1
tmp2 = i;

tmpSum = tmp1 + tmp2;

i = tmpSum;
```

0x03 小结

这篇文章，我们通过 `i++` 与 `++i` 字节码的不同讲述了两者的区别，希望能对你后续笔试遇到类似的题目有所帮助。

0x04 思考

留一道作业题给你，下面的代码输出是什么？你可以画出各阶段的过程图吗？

```
public static void foo() {
    int i = 0;
    i = ++i + i++ + i++ + i++;
    System.out.println("i=" + i);
}
```

欢迎你在留言区留言，和我一起讨论。