

## 本小节内容

## 补码讲解及内存实战演示

## 1 补码讲解及内存实战演示（组成常考选择题）

计算机的 CPU 无法做减法操作（硬件上没有减法器），只能做加法操作。CPU 中有一个逻辑单元叫加法器。计算机所做的减法，都是通过加法器将其变化为加法实现的。那么减法具体是如何通过加法实现的呢？实现 2-5 的方法是  $2+(-5)$ 。由于计算机只能存储 0 和 1，因此我们编写程序来查看计算机是如何存储 -5 的，5 的二进制数为 101，称为原码。计算机用补码表示 -5，补码是对原码取反后加 1 的结果，即计算机表示 -5 时会对 5 的二进制数（101）取反后加 1，如图 1 所示。-5 在内存中存储为 0xfffffb，因为对 5 取反后得 0xfffffa，加 1 后得 0xfffffb（由于是 X86 架构是小端存储，小端存储是低字节在前，即低字节在低地址，高字节在后，即高字节在高地址，fb 对于 0xfffffb 是最低的字节，因此 fb 在最前面，大端和小端相反），对其加 2 后得 0xfffffd，见图 2，它就是 k 的值。当最高位为 1（代表负数）时，要得到原码才能知道 0xfffffd 的值，即对其取反后加 1（当然，也可以减 1 后取反，结果是一样的）得到 3，所以其值为 -3。

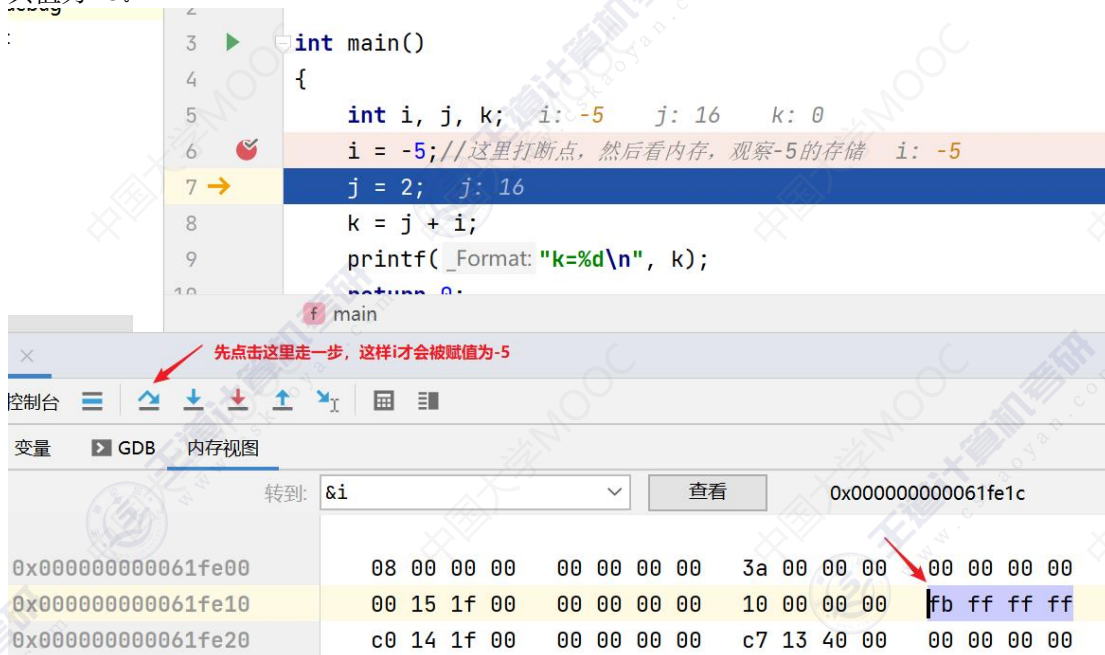


图 1 负数是以补码形式存储的

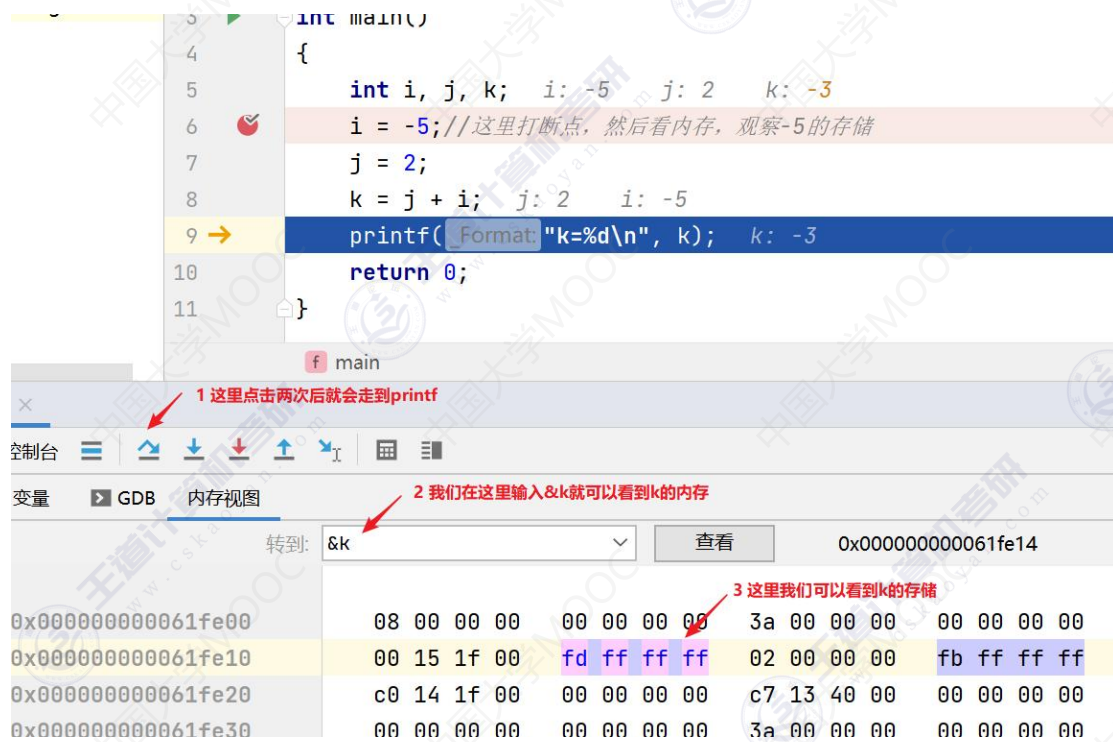


图 2 变量 k 的存储效果

注意，通过 8 位表示，-5 的补码 1111 1011，-5 的原码是 1000 0101 符号位是不动的，只有值的部分是 5。

反码（代码不好演示，考试考的概率低）：

反码是一种在计算机中数的机器码表示。对于单个数值（二进制的 0 和 1）而言，对其进行取反操作就是将 0 变为 1，1 变为 0。正数的反码和原码一样，负数的反码就是在原码的基础上符号位保持不变，其他位取反。

十进制	原码	反码
6	0000 0110	0000 0110
-3	1000 0011	1111 1100