

《计算机系统》理论课一位与信息&整数 1

课堂要点

本次课堂教学第一节课主要内容为：**位与信息、位运算操作、内存与指针及字符串**。通过介绍机器层面的信息存储与表达方式，让学生了解机器运算的基础知识。

1. **二进制因为**：a. 开关电路特性；b. 运算模式简单；c. 抗干扰性；d. 匹配布尔代数，而成为电子计算机使用的数制。（**仅需了解**）

二进制、十六进制、十进制三者之间的相互转换**需要熟练操作**（底权公式、幂次拼接等等），十六进制的**加减运算需要多多熟悉**，将来内存地址的运算会经常碰到。

2. **位运算与逻辑运算**：要注意位运算是将参数展开成位向量，然后**按位进行运算**，得到的结果依然是位向量；逻辑运算对于操作数**只判断是否非 0**，为 0 则假，非 0 则真。结果要么为 0x00（假），要么为 0x01（真）。

3. **移位运算**：

移位运算可以非常便捷地实现 2 的幂次的乘除法（但这不是移位的唯一用途）。逻辑左移和算术左移都是在移位后右侧补 0；逻辑右移，左侧补 0；**逻辑右移**，左侧补的是**符号位**（非负数补 0，负数补 1）。

4. **内存是字节的线性组合**，内存是依照字节编址的，也就是一个地址对应一个字节，内存中数据的存放遵循两大模式（大端法和小端法）。

小端法关键字：小、低、低（小端法就是低位字节放在数据低位上）；或者口诀：高高低低（高位数据去高地址，低位数据去低地址）

本次课堂教学第二节课主要内容为有符号数及其与无符号数之间的关系。通过介绍补码的定义、本质、求补方法、有符号数与无符号数关系映射等

1. **补码**：补码的定义为将首位作为符号位并变成负权重，从而实现对负数的表达；本质来说，补码是将最高位为 1 的二进制数“平移” 2^n 到负半轴，用来表

示与其有互补关系的负数。课堂举例： $w=4$, 1010 原本为+10, 变为补码则表示与+10 在 $2^4=16$ 范围内互补的-6。

2. **补码的值与取值范围**：使用 PPT 中的**补码值盒子**，或者利用**互补关系**，或者直接运用定义补码的**底权公式**，都可以很方便的求出补码的值。观察补码的取值范围可以发现，与原码和反码不同，补码只有一个 0，便于计算结果的确认。也因为单独一个 0 的存在，补码的范围是不对称的。 w 位有符号数（补码）的取值范围是： $-2^{w-1} \sim 2^{w-1}-1$ 。

3. **求补码的方法**：课堂上介绍了四种方法，1&2 适合大家理解补码本质，在 w 较小时也便于计算；3&4 比较直截了当，操作简单，适合 w 较大时，也是电路实现时经常采用的方式。

4. **相互映射和转换**。有符号数与无符号数相互间是一一映射的，映射的关键点，也是唯一的连接点，就是他们的二进制表达。一个二进制向量，可以分别作为有符号数和无符号数来解释，分别解释的这两个数之间，如果是非负数则二者相同（非负数的补码使其本身），如果其中一个为负数则二者之间相差 2^w 。在 C 语言中二者可以强制相互转换（具体方式见 PPT）。