

给C语言初学者 C语言中指针的作用

指针是C语言的一个重要特性。它提供了引用数据结构(包括数组)的元素的机制。与变量类似,指针也有两个方面:值和类型。它的值表示某个对象的位置,而它的类型表示那个位置上所存储对象的类型(比如整数或者浮点数)。

真正理解指针需要查看它们在机器级上的表示以及实现。这将是第3章的重点之一,3.10.1节将对其进行深入介绍。

2.1.1 十六进制表示法

一个字节由8位组成。在二进制表示法中,它的值域是 $00000000_2 \sim 11111111_2$ 。如果看成十进制整数,它的值域就是 $0_{10} \sim 255_{10}$ 。两种符号表示法对于描述位模式来说都不是非常方便。二进制表示法太冗长,而十进制表示法与位模式的互相转化很麻烦。替代的方法是,以16为基数,或者叫做十六进制(hexadecimal)数,来表示位模式。十六进制(简称为“hex”)使用数字‘0’~‘9’以及字符‘A’~‘F’来表示16个可能的值。图2-2展示了16个十六进制数字对应的十进制值和二进制值。用十六进制书写,一个字节的值域为 $00_{16} \sim FF_{16}$ 。

十六进制数字	0	1	2	3	4	5	6	7
十进制值	0	1	2	3	4	5	6	7
二进制值	0000	0001	0010	0011	0100	0101	0110	0111
十六进制数字	8	9	A	B	C	D	E	F
十进制值	8	9	10	11	12	13	14	15
二进制值	1000	1001	1010	1011	1100	1101	1110	1111

图2-2 十六进制表示法。每个十六进制数字都对16个值中的一个进行了编码

在C语言中,以0x或0X开头的数字常量被认为是十六进制的值。字符‘A’~‘F’既可以是大写,也可以是小写。例如,我们可以将数字 $FA1D37B_{16}$ 写作 $0xFA1D37B$,或者 $0xfald37b$,甚至是大小写混合,比如, $0xFa1D37b$ 。在本书中,我们将使用C表示法来表示十六进制值。

编写机器级程序的一个常见任务就是在位模式的十进制、二进制和十六进制表示之间人工转换。二进制和十六进制之间的转换比较简单直接,因为可以一次执行一个十六进制数字的转换。数字的转换可以参考如图2-2所示的表。一个简单的窍门是,记住十六进制数字A、C和F相应的十进制值。而对于把十六进制值B、D和E转换成十进制值,则可以通过计算它们与前三个值的相对关系来完成。

比如,假设给你一个数字 $0x173A4C$ 。可以通过展开每个十六进制数字,将它转换为二进制格式,如下所示:

十六进制	1	7	3	A	4	C
二进制	0001	0111	0011	1010	0100	1100

这样就得到了二进制表示 000101110011101001001100 。

反过来,如果给定一个二进制数字 1111001010110110011 ,可以通过首先把它分为每4位一组来转换为十六进制。不过要注意,如果位总数不是4的倍数,最左边的一组可以少于4位,前面用0补足。然后将每个4位组转换为相应的十六进制数字:

二进制	11	1100	1010	1101	1011	0011
十六进制	3	C	A	D	B	3



练习题 2.1 完成下面的数字转换: