

《现代C++编程实现》电子版: <https://leanpub.com/c01>

# 数和字符的表示

**YouTube频道:** [hwdong](#)

**博客:** [hwdong-net.github.io](https://hwdong-net.github.io)

# 数的表示

# 十进制

- 用10个不同的数字0、1、2、.....、9表示一个数。如329
- “逢十进一”，如11表示的是 $10+1$ ，整数329意思是“3个100”加上“2个10”加上“1个9”。
- 十进制整数可以表示成 $10^i$ 的多项式

$$329 = 3 \times 100 + 2 \times 10 + 9 = 3 \times 10^2 + 2 \times 10^1 + 9 \times 10^0$$

# 二进制

- 由于计算机硬件即晶体管只有“开”和“关”两种，也就是说一个晶体管开关只能区分2种不同情况，相当于只能表示0和1这2个数字。
- 采用“**逢2进1**”的方法，采用多个晶体管，即用多个0和1的排列来表示一个很大的数。
- 二进制数1011可以表示成 $2^i$ 的多项式：

$$\begin{aligned} 1011 &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 1 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1 = 11 \end{aligned}$$

# 二进制

- 1个二进制位经常也称为1比特(b)，而8个二进制位经常也称为1字节(B)。

表 1-1 部分 8 位二进制数对应的十进制数

二进制 ↵	十进制 ↵	↵
0000 0000 ↵	0 ↵	↵
0000 0001 ↵	1 ↵	↵
0001 0001 ↵	17 ↵	↵
0111 1111 ↵	127 ↵	↵
1000 0000 ↵	128 ↵	↵
1001 0001 ↵	145 ↵	↵
1111 1111 ↵	255 ↵	↵

# 二进制

- 在C++语言中，用0b开头的一串二进制表示一个二进制数。如二进制数1011用C++表示为：0b1011
- 如何将一个十进制整数转换为二进制整数？

$$n = (n/2) * 2 + n\%2 = \text{商} * 2 + \text{余数}$$

$$11 = (11/2) * 2 + 11\%2 = 5 * 2 + 1$$

← 二进制的末位

$$5 = (5/2) * 2 + 5\%2 = 2 * 2 + 1$$

← 二进制的倒数第2位

$$2 = (2/2) * 2 + 2\%2 = 1 * 2 + 0$$

← 二进制的倒数第3位

$$1 = (1/2) * 2 + 1\%2 = 0 * 2 + 1$$

← 二进制的倒数第4位

11对应的二进制数是1011

# 十六进制

- 当处理更大的二进制数会有些问题，二进制位太多了，不方便，如：

1011 0101 1101 1001 1110 0101

- 表示成十进制数11917797，只要8位数字。如要求将二进制的第17位设置为0。
- 一个解决方案是采用**十六进制**。即用16个不同的数来表示一个数字0、1、2、...、9和A(a)、B(b)、C(c)、D(d)、E(e)、F(f)表示一个数。

# 十六进制

表 1-2 十六进制、十进制和二进制的对应关系 ↵

十六进制 ↵	十进制 ↵	二进制 ↵
0 ↵	0 ↵	0000 ↵
1 ↵	1 ↵	0001 ↵
2 ↵	2 ↵	0010 ↵
3 ↵	3 ↵	0011 ↵
4 ↵	4 ↵	0100 ↵
5 ↵	5 ↵	0101 ↵
6 ↵	6 ↵	0110 ↵
7 ↵	7 ↵	0111 ↵
8 ↵	8 ↵	1000 ↵
9 ↵	9 ↵	1001 ↵
A 或 a ↵	10 ↵	1010 ↵
B 或 b ↵	11 ↵	1011 ↵
C 或 c ↵	12 ↵	1100 ↵
D 或 d ↵	13 ↵	1101 ↵
E 或 e ↵	14 ↵	1110 ↵
F 或 f ↵	15 ↵	1111 ↵



# 十六进制

- 因为一个十六进制数对应4个二进制数，所以你可以将任何二进制数按照4个一组的方式以十六进制形式表示。如：

1011 0101 1101 1001 1110 0101

- 对应的十六进制表示法为：

b 5 d 9 e 5

- 十六进制采用如下方式计算出对应的十进制的值：

$$b \times 16^5 + 5 \times 16^4 + d \times 16^3 + 9 \times 16^2 + e \times 16^1 + 5 \times 16^0$$

# 十六进制

- C++中以0x或0X开头表示一个十六进制数：  
0xb5d9e5

# 八进制

- 即用0、1、2、3、4、5、6、7表示一个数。
- C++中以0开头表示一个十六进制数：

0354745

# 以不同进制格式输出

- `std::dec`、`std::hex`、`std::oct`操作符以十进制、十六进制、八进制形式输出。

```
std::cout << "十进制: " <<
std::dec << 329 << std::endl
<< "十六进制: "
<< std::hex << 329 << std::endl
<< "八进制: "
<< std::oct << 329 << std::endl;

std::cout << "十进制: " <<
std::dec << 0b101101011101100111100101 << std::endl
<< "十六进制: "
<< std::hex << 0b101101011101100111100101 << std::endl
<< "八进制: "
<< std::oct << 0b101101011101100111100101 << std::endl;
```

# 字符的表示

# ASCII码

- 在计算机中，各种字符如大小写的英文字母、数字(0、1、2、...、9)、一些特殊字符（如#、/、换行符、制表符等键盘上可见字符）也都是以二进制串表示的。
- 整数值为35的二进制串表示字符#。不同的二进制串（对应一个整数）表示不同的字符完全取决于你怎么解释它们。
- ASCII码：开始用7位后来用8位二进制表示一个字符。如英文、德文、法文字符。

<http://ascii.911cha.com/>

# 统一字符编码 UCS

- Universal Character Set (UCS) : 用32位二进制整数表示一个字符，包括汉字、表情符号、图形符号等。每个字符都对应一个确定整数值表示的编码点。

# Unicode标准

- 不仅定义了一个字符集及其编码点（其中字符的编码点等同于UCS的对应字符的编码点），还定义了多种表示编码点的**编码方式**。
- 最广泛使用的Unicode编码方式包括**UTF-8、UTF-16、UTF-32**。其中每个编码方式都可以表示Unicode字符集中的所有字符。



# Unicode标准中的码点

- Unicode 码点是一个整数（最多 21 位），唯一标识一个字符。
- Unicode 规定的码点范围是：从 **U+0000** 到 **U+10FFFF**（即十进制的 0 到 1,114,111）。
- 这意味着码点最多占 **21 位二进制**（因为  $0x10FFFF = 1,114,111 = 100001111111111111111_2$ ）
- 码点不是存储格式，计算机存储字符时，会使用特定的编码方案（如 UTF-8），而不是直接存储 Unicode 码点的十六进制值。

# Unicode标准中的编码方式

- UTF-8表示一个字符采用的是变长的字节序列(即用1~4字节表示一个字符), 其中ASCII字符的编码使用1字节表示, 和对应的ASCII编码是一样的。大多数网页的文本都采用UTF-8编码方式.
- UTF-16用1个或2个16位编码表示一个字符。UTF-16包含了UTF-8。
- UTF-32最简单, 用一个32位编码表示所有的字符。

# Unicode编码点和UTF-8编码的关系

表 1-3 Unicode 编码点和 UTF-8 编码关系 ↵

Unicode 编码点（十六进制形式） ↵	UTF-8 编码（二进制形式） ↵
0000 0000-0000 007F ↵	0xxxxxxx ↵
0000 0080-0000 07FF ↵	110xxxxx 10xxxxxx ↵
0000 0800-0000 FFFF ↵	1110xxxx 10xxxxxx 10xxxxxx ↵
0001 0000-0010 FFFF ↵	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx ↵

# UniCode编码点和UTF-8编码的关系

表 1-3 UniCode 编码点和 UTF-8 编码关系 ↵

UniCode 编码点（十六进制形式） ↵	UTF-8 编码（二进制形式） ↵
0000 0000-0000 007F ↵	0xxxxxxx ↵
0000 0080-0000 07FF ↵	110xxxxx 10xxxxxx ↵
0000 0800-0000 FFFF ↵	1110xxxx 10xxxxxx 10xxxxxx ↵
0001 0000-0010 FFFF ↵	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx ↵

表 1-4 字符 A 和汉字'中'的 UniCode 编码点及 UTF-8 编码 ↵

字 符 ↵	ASCII ↵	UniCode（十六进 制） ↵	UniCode（二进制） ↵	UTF-8 ↵
A ↵	01000001 ↵	0041 ↵	00000000 01000001 ↵	01000001 ↵
中 ↵	无 ↵	4E2D ↵	01001110 00101101 ↵	11100100 10111000 10101101 ↵

# 练习

- 编写程序在十进制和十六进制或二进制之间进行转换

9. 将十进制整数 765431 转换为十六进制，在十六进制形式下，如何将该数的二进制的第 17 位设置为 1？写出对应的二进制形式。
10. 请在网上查询一下换行符、回车符、制表符和汉字‘汉’的 Unicode 编码点（值），然后根据表 1-4 写出其对应的 UTF-8 的编码。

# 关注我

博客: [hwdong-net.github.io](https://hwdong-net.github.io)

Youtube频道



**hwdong**

@hwdong · 5.01K subscribers · 558 videos

博客: <https://hwdong-net.github.io> >

[youtube.com/c/4kRealSound](https://youtube.com/c/4kRealSound) and 4 more links