

存储单位

位

二进制位简称“位”，一般用 1 或 0 表示，一个二进制位包含的信息量称为一比特，简记为 b。计算机中的 CPU 位数指的是 CPU 一次能处理的最大位数。

字节

字节是计算机可寻址的最小单位，每个字节有 8 个二进制位，其中最右边的一位为最低位，最左边的一位为最高位，每个二进制位的值不是 0 就是 1。

$$1 \text{ 字节 (Byte)} = 8\text{b} = 2^0 \text{ 字节}$$

$$1 \text{ 千字节 (Kb)} = 1024\text{Byte} = 2^{10} \text{ 字节}$$

$$1 \text{ 兆字节 (Mb)} = 1024\text{Kb} = 2^{20} \text{ 字节}$$

$$1 \text{ 吉字节 (Gb)} = 1024\text{Mb} = 2^{30} \text{ 字节}$$

$$1 \text{ 太字节 (Tb)} = 1024\text{Gb} = 2^{40} \text{ 字节}$$

小提示：“字节”是计算机中数据存储的最小单位。

小提示：1 字节能够存储的最大数值为 255。

1	1	1	1	1	1	1	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

$$= 255$$

课后练习

1、[2016 年普及组初赛 2] 如果 256 种颜色用二进制编码来表示，至少需要（ ）位。

- A 6 B 7 C 8 D 9

2、[2016 年提高组初赛 9] 某计算机的 CPU 和内存之间的地址总线宽度是 32 位 (bit)，这台计算机最多可以使用（ ）的内存。

- A 2GB B 4GB C 8GB D 16GB

3、[2014 年普及组初赛 12] 下列几个 32 位 IP 地址中，书写错误的是（ ）。

- A 10.0.0.1
B 192.168.0.1
C 256.256.129.1
D 162.105.135.27

4、[2008 年普及组初赛 17] 在 32×32 点阵的“字库”中，汉字“北”与“京”的字模占用字节数之和是（ ）。

- A 512 B 128 C 256 D 384

5、[2010 年普及组初赛 2、2010 年提高组初赛 2] 一个字节 (byte) 由（ ）个二进制位组成。

- A 8 B 16 C 32 D 以上都有可能

6、[2011 年普及组初赛 3] 一片容量为 8GB 的 SD 卡能存储大约（ ）张大小为 2MB 的数码照片。

- A 1600 B 2000 C 4000 D 16000

7、[2020 CSP-S1 提高级初赛 3] 现有一段 8 分钟的视频文件，它的播放速度是每秒 24 帧图像，每帧图像是一幅分辨率为 2048×1024 像素的 32 位真彩色图像。请问要存储这段原始无压缩视频，需要多大的存储空间？（ ）

- A 30GB B 90GB C 150GB D 450GB

8、[2014 年普及组初赛 2] 1TB 表示的字节数是（ ）。

- A 2 的 10 次方 B 2 的 20 次方 C 2 的 30 次方 D 2 的 40 次方

参考答案：

1、C 解析：一位二进制可以表示两种颜色。例如 0 为黑色，1 为白色。所以 256 种颜色，只需要 8 位， $2^8=256$ 。

2、B 解析： $2^{32}=4G$ 。

3、C 解析：32 位 IP 地址，每一位最大的值为 255

4、C 解析：32 点阵即为 32 位，等于 4 字节。 $32 * 32 / 8 * 2 = 256$

5、A

6、C 解析： $8G = 8 * 1024K = 8192M / 2M = 4096$ 张

7、B 解析： $2048 * 1024 * 32 / 8 * 24 * 60 * 8 / 1024 = 90G$

8、D 解析：

1 字节(Byte) = 8bit = 20 字节

1 千字节(Kb) = 1024b = 210 字节

1 兆字节(Mb) = 1024Kb = 220 字节

1 吉字节(Gb) = 1024Mb = 230 字节

1 太字节(Tb) = 1024Gb = 240 字节

机器数

一个数在计算机中的二进制表示形式，叫做这个数的**机器数**。机器数是带符号的，在计算机用一个数的最高位存放符号，正数为0，负数为1。

比如，十进制中的数 +3，计算机字长为8位，转换成二进制就是 00000011。如果是 -3，就是 10000011。

那么，这里的 00000011 和 10000011 就是机器数。

真值

因为第一位是符号位，所以机器数的形式值就不等于真正的数值。例如上面的有符号数 10000011，其最高位1代表负，其真正数值是 -3 而不是形式值 131

(10000011 转换成十进制等于 131)。所以，为区别起见，将带符号位的机器数对应的真正数值称为机器数的真值。

例：0000 0001 的真值 = +000 0001 = +1，1000 0001 的真值 = -000 0001 = -1

原码

原码就是符号位加上真值的绝对值，即用第一位表示符号，其余位表示值。

优点：简单直观；例如，我们用8位二进制表示一个数，+11的原码为 00001011，-11的原码就是 10001011

缺点：原码不能直接参加运算，可能会出错。例如数学上， $1+(-1)=0$ ，而在二进制中原码 $00000001+10000001=10000010$ ，换算成十进制为-2。显然出错了。

反码

反码通常是用来由原码求补码或者由补码求原码的过渡码

正数的反码是其本身

负数的反码是在其原码的基础上，符号位不变，其余各个按位取反。

$[+1] = [00000001]$ 原 = $[00000001]$ 反

$[-1] = [10000001]$ 原 = $[11111110]$ 反

补码

正数的补码就是其本身

负数的补码是在其原码的基础上，符号位不变，其余各位取反，最后+1。（即在反码的基础上+1）

$$[+1] = [00000001]_{\text{原}} = [00000001]_{\text{反}} = [00000001]_{\text{补}}$$

$$[-1] = [10000001]_{\text{原}} = [11111110]_{\text{反}} = [11111111]_{\text{补}}$$

机器数的意义

思路：用加法代替减法，例如 $1 + (-1) = 1 - 1 = 0$ ；

$$1 - 1 = 1 + (-1) = [00000001]_{\text{原}} + [10000001]_{\text{原}} = [10000010]_{\text{原}} = -2$$

$$1 - 1 = 1 + (-1) = [0000\ 0001]_{\text{原}} + [1000\ 0001]_{\text{原}} = [0000\ 0001]_{\text{反}} + [1111\ 1110]_{\text{反}} = [1111\ 1111]_{\text{反}} = [1000\ 0000]_{\text{原}} = -0。$$

$$1 - 1 = 1 + (-1) = [0000\ 0001]_{\text{原}} + [1000\ 0001]_{\text{原}} = [0000\ 0001]_{\text{补}} + [1111\ 1111]_{\text{补}} = [0000\ 0000]_{\text{补}} = [0000\ 0000]_{\text{原}}$$

数学原理：同余，不再扩展。

小提示：

- 1、算术运算对象与结果在机器内部都是以补码的形式存储的，只是在输出时被还原成原码。
- 2、所有的数学运算在机器内部都是以补码的形式进行计算的。
- 3、由于计算机运算器的特性，所以最高位进位时，不会溢出，会被直接舍弃。原理：场运算，不扩展。
- 4、原码和反码能够表示的范围是-127 到 127，共 255 个。
- 5、补码能够表示的范围是-128 到 127 共 256 个。因为补码中没有-0 的概念，所以 10000000 就被表达为-128。

课后练习

1、[2009 年提高组初赛 4]在字长为 16 位的系统环境下，一个 16 位带符号整数的二进制补码为 111111111101101。其对应的十进制整数应该是：

- A 19 B -19 C 18 D -18

2、[2010 年普及组初赛 11]一个字长为 8 位的整数的补码是 11111001，则它的原码是（ ）。

- A 0000111 B 01111001 C 11111001 D 1000111

3、[2010 年提高组初赛 14]在整数的补码表示法中，以下说法正确的是（ ）。

- A 只有负整数的编码最高位为 1
B 在编码的位数确定后，所能表示的最小整数和最大整数的绝对值相同
C 整数 0 只有一个唯一的编码
D 两个用补码表示的数相加时，若在最高位产生进位，则表示运算溢出

4、[2015 年普及组初赛 4]在计算机内部用来传送、存贮、加工处理的数据或指令都是以（ ）形式进行的。

- A 二进制码 B 八进制码 C 十进制码 D 智能拼音码

参考答案

1、B 解析：先求反码 111111111101100，再求原码 10000000010011 = -19

2、D 解析：先求反码 11111000，再求原码 1000111

3、AC 解析：A 定义。B 补码中最小整数为-128，最大整数位 127。C 正确 D，不溢出，场计算原理。

4、A

位运算

按位与

按位与运算符“&”是双目运算符。其功能是参与运算的两数各对应的二进位相与。只有对应的两个二进位都为1时，结果位才为1。

与计算图解

数字A	0	1	1	0	1
数字B	1	0	1	1	0
同1为1	1	0	1	0	0

示例代码：

```
int a = 3, b = 10, z;
```

```
z = a & b;
```

按位或

按位或运算符“|”是双目运算符。其功能是参与运算的两数各对应的二进位相或。只要对应的二个二进位有一个为1时，结果位就为1。

或计算示例

数字A	0	1	1	0	1
数字B	1	0	1	0	0
有1为1	1	1	1	0	1

示例代码：

```
int a = 3, b = 10, z;
```

```
z = a | b;
```

按位非

按位非运算符“~”是单目运算符。作用是将每位二进制取反：1 变 0；0 变 1。

非计算示例

数字A	0	1	1	0	1
全部取反	1	0	0	1	0

示例代码：

```
int a = ~1;
```

按位异或

按位异或运算符“^”是双目运算符。其功能是参与运算的两数各对应的二进制位相异或。只要对应的二个二进制数值相同时，结果位就为 0，否则为 1。

异或计算示例

数字A	0	1	1	0	1
数字B	1	0	1	0	0
相同为0	1	1	0	0	1

示例代码：

```
int a = 1;
int b = -1;
int c = a^b;
```

小提示：以上所有计算，均已数值的补码形式参与。

左移

左移运算符是指将一个数的各二进制位全部左移若干位，移动的位数由右操作数指定，右操作数必须是非负值，其右边空出的位用 0 填补，高位左移溢出则舍弃该高位。

左移运算示例

舍弃	原始位置			
0	1	1	0	1
1	0	1	0	0
新位置			使用0进行填充	

示例代码：

```
int a = 1 << 1; //输出 2
```

小提示：

- 1、左移一位等于当前数值乘以 2，左移的效率比计算乘法要高。
- 2、左移可能会改变数值的符号。

有符号右移

右移运算符是指将一个数的各二进制位全部右移若干位，移动的位数由右操作数指定，右操作数必须是非负值。保证高位符号位不变，其左边空出的位用 1 填补（因为运算采用的是反码，所以左边都是 1），低位右移溢出则舍弃该位。

有符号右移运算示例

原始位置			舍弃	
0	1	1	0	1
0	1	0	1	1
符号位不变 高位补1			新位置	

示例代码：

```
int a = 4 >> 1; //输出 2
```

小提示：右移一位等于当前数值除以 2，右移的效率比计算除法要高。

无符号右移

无符号右移“>>>”与有符号右移的区别就是最大位始终补 0。

无符号右移运算示例

原始位置			舍弃	
0	1	1	0	1
0	0	0	1	1
高位始终补0			新位置	

小提示：c++不直接提供无符号右移

重点：全部的位运算均是以数值的补码形式参与计算，非原码！

课后练习

1、[2016 提高 3] 二进制数 00101100 和 01010101 异或的结果是 ()。

A 00101000 B 01111001 C 01000100 D 00111000

参考答案

1、B 解析：异或的定义是按位比较，如果二者相同则为 0，其他为 1。

辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

