**浮点数在内存中的存储方式**

原创 2016年11月05日 17:53:49

* 标签：
* [c语言](http://so.csdn.net/so/search/s.do?q=c%E8%AF%AD%E8%A8%80&t=blog) /
* [浮点数存储方式](http://so.csdn.net/so/search/s.do?q=%E6%B5%AE%E7%82%B9%E6%95%B0%E5%AD%98%E5%82%A8%E6%96%B9%E5%BC%8F&t=blog)
* 2461

浮点数类型包括float、double、long double

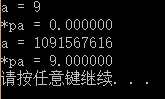
在这里以float为例。

先看一段代码：

**[cpp]** [view plain](https://blog.csdn.net/d_leo/article/details/53046721) [copy](https://blog.csdn.net/d_leo/article/details/53046721)

1. <span style="font-size:14px;">#include<stdio.h>
2. #include<windows.h>
3. **int** main()
4. {
5. **int** a = 9;
6. **float** \*pa = (**float** \*)&a;
7. printf("a = %d\n", a);
8. printf("\*pa = %f\n", \*pa);
9. \*pa = 9.0;
10. printf("a = %d\n", a);
11. printf("\*pa = %f\n", \*pa);
12. system("pause");
13. **return** 0;
14. }</span>

它的运行结果如下：

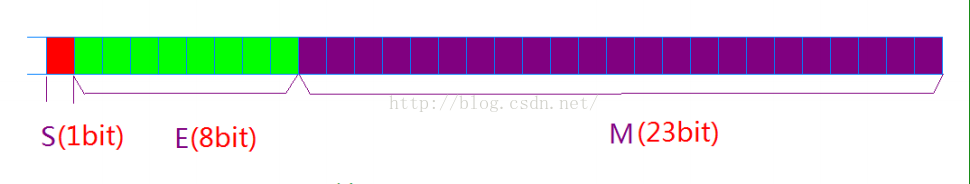


为什么会有这样的结果呢？简单分析一下：  
对于9和9.0在内存中肯定都以补码的形式存在，因为int和float对于这串补码的处理方式不同所以才会得到不一样的结果。  
而对于float到底是怎样在内存中存储的呢？这就是我们讨论的重点!

根据国际标准IEEE（电气和电子工程协会）规定，任何一个浮点数NUM的二进制数可以写为：  
NUM = (-1)^S\*M\*2^E;//(S表示符号，E表示阶乘，M表示有效数字)  
①当S为0时，表示一个正数；当S为1时，表示一个负数  
②M表示有效数字，1<= M <2  
③2^E表示指数  
比如十进制的3.0，二进制就是0011.0 就可以写成（-1）^0\*1.1\*2^1   
在比如十进制的-3.0，二进制就是-0011.0 就可以写成（-1）^1\*1.1\*2^1   
而规定float类型有一个符号位（S），有8个指数位（E），和23个有效数字位（M）

     double类型有一个符号位（S），有11个指数位（E），和52个有效数字位（M）

以float类型为例：

  
IEEE对于（有效数字）M和（指数）E有特殊的规定： （以float为例）  
1.因为M的值一定是1<= M <2，所以它绝对可以写成1.xxxxxxx的形式，所以规定M在存储时舍去第一个1，只存储小数点之后的数字。这样做节省了空间，以float类型为例，就可以保存23位小数信息，加上舍去的1就可以用23位来表示24个有效的信息。  
2.对于E（指数）E是一个无符号整数所以E的取值范围为（0~255），但是在计数中指数是可以为负的，所以规定在存入E时，在它原本的值上加上中间数（127），在使用时减去中间数（127），这样E的真正取值范围就成了（-127~128）。

对于E还分为三种情况：

①E不全为0，不全为1:

这时就用正常的计算规则，E的真实值就是E的字面值减去127（中间值)，M的值要加上最前面的省去的1。

②E全为0

这时指数E等于1-127为真实值，M不在加上舍去的1，而是还原为0.xxxxxxxx小数。这样为了表示0，和一些很小的整数。

所以在进行浮点数与0的比较时，要注意。

③E全为1

当M全为0时，表示±无穷大（取决于符号位）；当M不全为1时，表示这数不是一个数（NaN）

在看刚开始的题目：

①int a = 9;因为a是int类型的，所以他在内存中以补码的形式储存：

00000000 00000000 00000000 00001001

而\*pa却是float类型的所以，当\*pa读这块内存的值时，它通过浮点数的形式读取

0     0000000 0     0000000 00000000 00001001

—   —————    ———————————————

S           E                                M

\*pa = （-1）^0 \* (0.0000000 00000000 00001001) \* 2^(1-127)  这个数是一个很小的数，用十进制小数表示就是 0.000000

②\*pa = 9.0;因为\*pa是float类型的，所以9可以写为（1001）= (-1)^0 \* (1.001) \* 2^(3)

所以：

S = 0；M = 001000…… E = 3 + 127 = 130

0    1000001 0        0010000 00000000 00000000

—   —————    ———————————————

S           E                                M

而把这个二进制数还原为十进制数就为1091567616就是a的值

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/D\_leo/article/details/53046721