<https://www.cnblogs.com/shirishiqi/p/5392854.html>

最近在做一个C的嵌入式项目，发现在C语言中用printf()函数打印字符型变量时，如果想采用"%x"的格式将字符型变量值以十六进制形式打印出来，会出现一个小问题，如下：

C代码  收藏代码

1. char buf[10] = {0};
2. buf[0] = 0xbf;
3. printf("%2x\n\n\n", buf[0]);            /\*在终端将会显示成：ffffffbf\*/
5. buf[1] = 0x7f;
6. printf("%2x\n\n\n", buf[1]);            /\*在终端将会显示成：7f\*/

 为什么buf[0]不像buf[1]显示的那样，显示2个十六进制数bf呢？

研究发现，只要字符型变量值的二进制第一位是1，就会如buf[0]所显示的那样，出现6个f。

而格式输出函数printf中会对所输出的变量做有符号/无符号型的判断。如果是有符号型变量，且该值二进制首位为1（如我们定义的是char buf[100]，且buf[0]中为0xbf），则会按照补码形式前面全置为1，也就是全为f。（因为终端显示的是8个十六机制数字；另外，我用Tobor C编译，因为Tobor C中int是2bytes，终端显示就成4个十六进制数字了：ffbf，因此我猜测，printf函数显示之前先将有符号型值转换成了一个4bytes的int。当然，这是函数内部的处理，跟我们所说的话题关系不大。）。

按照上面述说的，如果把buf声明为unsigned char型，则显示会变成：bf

而且，如果程序需要讲一个有符号型的变量以十六进制形式输出，且只显示两位，可以强制类型转换为无符号型变量，如上面的buf[0]，可以转换成：（unsigned char）buf[0]

此外，如果有人这样来测试这个问题，

C代码  收藏代码

1. printf("%2x\n\n\n", 0xbf);               /\*在终端将会显示成：bf\*/

 为什么这边没有6个f呢？因为，printf()函数第二个参数直接以int型来接收，int型变量4bytes，二进制首位自然为0，因此不会显示多个f。