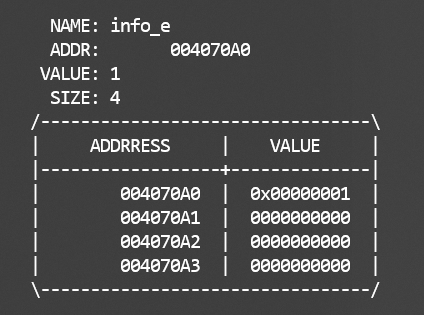
**2.55** 在我的PC上，其字节使用方式如下图所示。



这是一个int型变量，值为1。将其对应的四个字节的内容全部输出后，发现仅有地址最小的一个字节的值不为0，其值等于该变量的值。因此我的PC的字节使用方式为：从低位到高位，地址按从小到达的顺序依次分配，即这是一台小端机。

在我的Mac上，实验结果与PC上相同，不再赘述。

**2.59** 见附件中的2-59.c。

**2.63** 见附件中的2-63.c。

**2.67** (A) 右移后的位数大于int型字节宽度；(B) (C)见附件中的2-67.c。

**2.71** (A) 错误：unsigned右移是逻辑右移。而按照需求（获得int），这里应使用算术右移。

(B) 见附件中的2-71.c。

**2.75** 见附件中的2-75.c。

**2.79** 见附件中的2-79.c。

**2.83** 程序实现见附件中的2-83.c。

(A) *Y* = *y* / (2*k*– 1)

(B) 5/7, 2/5, 19/63

**2.87**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 描述 | Hex | M | E | V | D |
| -0 | 8000 | 0 | -14 | -0 | -0.0 |
| 最小的>2的值 | 4001 | 1025/1024 | 1 | 1025×2-9 | 2.001953125 |
| 512 | C000 | 0 | 9 | 512 | 512.0 |
| 最大的非规格化数 | 03FF | 1023/1024 | -14 | 1023×2-24 | 0.0000609755516 |
| -∞ | FC00 | - | - | -∞ | -∞ |
| 十六进制为3BB0的数 | 3BB0 | 123/64 | -1 | 123×2-7 | 0.9609375 |

**2.91** 程序验证见2-91.c。

(A) 11 . 0010 0100 0011 1111 0110 11

(B) 11. 001 001 001 001 ...

(C) 从小数点后第9位起不同。

**2.95** 见附件中的2-95.c。

**另：老师，我Lab1上交的实验报告漏填了一个空，是第五章的“HelloWin.c在Linux下用gcc缺省模式编译后运行结果为：”（如下图），此空应填“**Hello 1183710106����**”。如果可以，希望麻烦您重新判一下分，谢谢您！**

