



南京大學  
NANJING UNIVERSITY



# 数据存储时的字节排列

南京大学

计算机科学与技术系

袁春风

email: [cfyuan@nju.edu.cn](mailto:cfyuan@nju.edu.cn)

2015.6

# 数据的存储和排列顺序

- 80年代开始，几乎所有通用计算机都采用**字节编址**
- 在高级语言中声明的基本数据类型有char、short、int、long、long long、float、double、long double等各种**不同长度**数据
- 一个基本数据可能会占用多个存储单元

– 例如，若int型变量 $x = -10$ ， $x$ 的存放地址为100，其机器数为  
FFFFFFFF6H，占4个单元

$$-10 = -1010B$$

$$[-10]_{\text{补}} = \text{FFFFFFFF6H}$$

- 需要考虑以下问题：

– 变量的地址是其最大地址还是最小地址？

**最小地址，即 $x$ 存放在100#~103#！**

– 多个字节在存储单元中存放的顺序如何？

**大端方式/小端方式**

103#	FF	100#
102#	FF	101#
101#	FF	102#
100#	F6	103#

# 数据的存储和排列顺序

若  $\text{int } i = -65535$  , 存放在100号单元 ( 占 100 ~ 103 ) , 则用 “取数” 指令访问100号单元取出  $i$  时 , 必须清楚  $i$  的4个字节是如何存放的。

$$65535 = 2^{16} - 1$$

$$[-65535]_{\text{补}} = \text{FFFF0001H}$$

变量  $i$

FF	FF	00	01
103	102	101	100
MSB			LSB
100	101	102	103

小端 ( little endian )

大端 ( big endian )

大端方式 ( Big Endian ) : MSB所在的地址是数的地址

e.g. IBM 360/370, Motorola 68k, MIPS, Sparc, HP PA

小端方式 ( Little Endian ) : LSB所在的地址是数的地址

e.g. Intel 80x86, DEC VAX

有些机器两种方式都支持 , 可通过特定控制位来设定采用哪种方式。

# 检测系统的字节顺序

- union的存放顺序是所有成员从低地址开始，利用该特性可测试CPU的大/小端方式。

```
#include <stdio.h>
void main()
{
    union NUM
    {
        int a;
        char b;
    } num;
    num.a = 0x12345678;
    if(num.b == 0x12)
        printf("Big Endian\n");
    else
        printf("Little Endian\n");
    printf("num.b = 0x%X\n", num.b);
}
```

a 和 b 共用同一个空间

100    101    102    103

12	34	56	78	大端 小端
78	56	34	12	

猜测在IA-32  
上的打印结果

Little Endian num.b = 0x78
-------------------------------

# 大端/小端方式举例

假定小端方式机器中某条指令的地址为1000

该指令的汇编形式为：`mov AX, 0x12345(BX)`

其中操作码mov为40H，寄存器AX和BX的编号分别为0001B和0010B，立即数占32位，则存放顺序为：



若在大端机器上，则存放顺序如何？



00	1005	45
01	1004	23
23	1003	01
45	1002	00
12	1001	12
40	1000	40
地址		

只需要考虑指令中立即数的顺序！

# 大端/小端方式举例

- 以下是一个由反汇编器生成的一行针对IA-32处理器的机器级代码表示文本：

80483d2: 89 85 a0 fe ff ff mov %eax, 0xfffffea0(%ebp)

其中，80483d2是十六进制表示的指令地址

89 85 后四位是立即数 a0 fe ff ff 是机器指令

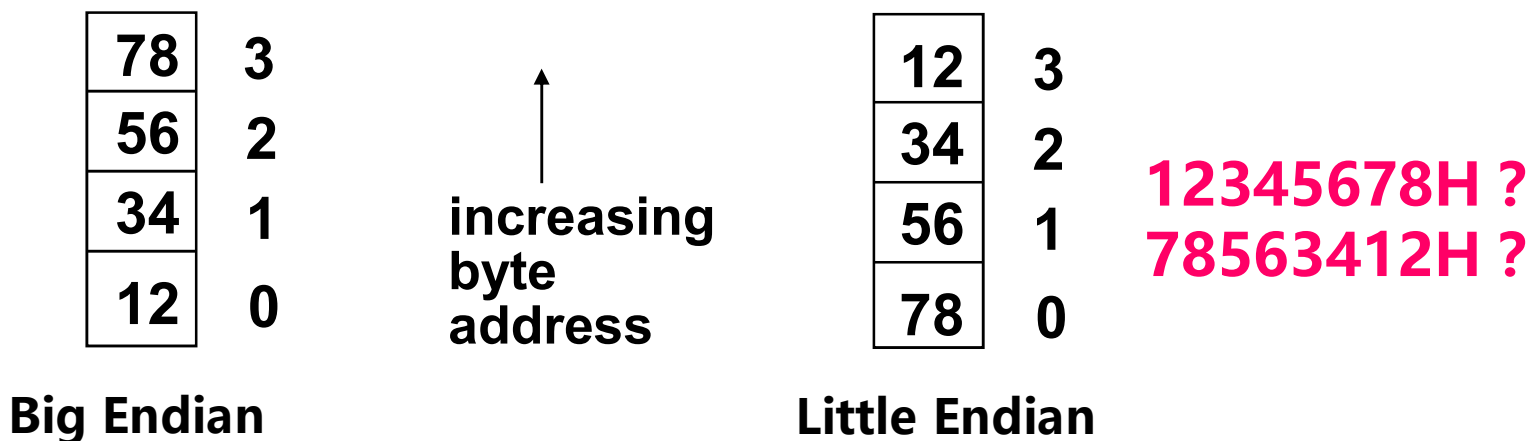
mov %eax, 0xfffffea0(%ebp) 是对应的汇编指令

0xfffffea0是立即数

请问：立即数0xfffffea0的值和所存放地址分别是多少？  
IA-32是大端还是小端方式？

- 立即数0xfffffea0所存放的地址为0x80483d4；
- 立即数0xfffffea0的值为-10110000B=-176；
- IA-32采用的是小端方式！

# 字节交换问题



上述存放在0号单元的数据（字）是什么？

存放方式不同的机器间程序移植或数据通信时，会发生什么问题？

- ◆ 每个系统内部是一致的，但在系统间通信时可能会发生问题！
- ◆ 因为顺序不同，需要进行顺序转换

音、视频和图像等文件格式或处理程序都涉及到字节顺序问题

**ex. Little endian: GIF, PC Paintbrush, Microsoft RTF, etc**

**Big endian: Adobe Photoshop, JPEG, MacPaint, etc**