**大端、小端**：指的是CPU存储数据的方式。大端的话存储的顺序是由低地址到高地址存储，即由小到大，所以叫大端；小端的话则与大端恰恰相反，是从高地址到底地址存储。

例： 比如0x01020304这个数，在Windows和linux系统下的布局是：

[01][02][03][04] 左边是高地址，右边是低地址

而其在Unix系统下的布局则是：

[04][03][02][01] 左边是高地址，右边是低地址

**网络字节序**：就是网络传输的字节顺序，这取决于软件开始时通讯双方的协议规定，默认是大端序

**主机字节序**：与具体机型有关

**位序**：字节中的位的顺序，开发时，我们不关心位序，由编译器和CPU来决定这些事情

**算数运算与内存操作运算**：算数运算并不影响字节序，而内存操作运算的过程中必须考虑字节序，不然很容易出错

**浮点数的字节序**：对于浮点数，不要直接传内存值，最好是按照字符串传输

**IP地址的字节序**：IP地址已经是网络字节序了，我们只需要做算数赋值运算即可。

**字节序转换函数**：OS一般都提供htons、htonl、ntohs、ntohl这四个字节序操作函数，在WIN下这四个函数会改变所操作数的数值，但是在UNIX下就不会改变数据的算数值，UNIX下这些函数是空操作。这个差异也是UNIX服务器上的程序性能会高于WIN的一个小原因。