

字节序

维基百科，自由的百科全书

字节顺序，又称**端序**，**尾序**（英语： Endianness）。在计算机科学领域中，是跨越多字节的程序对象的存储规则。

目录

- 1 简介
- 2 端（endian）的起源
- 3 字节顺序
 - 3.1 大端序
 - 3.2 小端序
 - 3.3 混合序
- 4 处理器体系
- 5 网络序
- 6 位序
 - 6.1 小端序（先传低位）的串行协议
 - 6.2 大端序（先传高位）的串行协议
- 7 参见
- 8 外部链接
- 9 参考

简介

在几乎所有的机器上，多字节对象都被存储为连续的字节序列。例如在C语言中，一个类型为int的变量x地址为0x100，那么其对应地址表达式&x的值为0x100。且x的四个字节将被存储在存储器的0x100, 0x101, 0x102, 0x103位置。^[1]

而存储地址内的排列则有两个通用规则。一个多位的整数将按照其存储地址的最低或最高字节排列。如果最低有效字节在最高有效字节的前面，则称小端序；反之则称大端序。在网络应用中，字节序是一个必须被考虑的因素，因为不同机器类型可能采用不同标准的字节序，所以均按照网络标准转化。

例如假设上述变量x类型为int，位于地址0x100处，它的十六进制为0x01234567，地址范围为0x100~0x103字节，其内部排列顺序依赖于机器的类型。大端法从首位开始将是：0x100: 01, 0x101: 23,...。而小端法将是：0x100: 67, 0x101: 45,...。

端（endian）的起源

“endian”一词来源于乔纳森·斯威夫特的小说格列佛游记。小说中，小人国为水煮蛋该从大的一端（Big-End）剥开还是小的一端（Little-End）剥开而争论，争论的双方分别被称为“大端派”和“小端派”。以下是1726年关于大小端之争历史的描述：

“我下面要告诉你的是，Lilliput和Blefuscu这两大强国在过去36个月里一直在苦战。战争开始是由于以下的原因：我们都认为，吃鸡蛋前，原始的方法是打破鸡蛋较大的一端，可是当今皇帝的祖父小时候吃鸡蛋，一次按古法打鸡蛋时碰巧将一个手指弄破了。因此他的父亲，当时的皇帝，就下了一道敕令，命令全体臣民吃鸡蛋时打破鸡蛋较小的一端，违令者重罚。老百姓们对这项命令极其反感。历史告诉我们，由此曾经发生过6次叛乱，其中一个皇帝送了命，另一个丢了王位。这些叛乱大多都是由Blefuscu的国王大臣们煽动起来的。叛乱平息后，流亡的人总是逃到那个帝国去寻求避难。据估计，先后几次有11000人情愿受死也不肯去打破鸡蛋较小的一端。关于这一争端，曾出版过几百本大部著作，不过大端派的书一直是受禁的，法律也规定该派任何人不得做官。”^[2]

1980年，Danny Cohen，一位网络协议的早期开发者，在其著名的论文“On Holy Wars and a Plea for Peace”中，为平息一场关于字节该以什么样的顺序传送的争论，而第一次引用了该词。^[3]

字节顺序

在哪种字节顺序更合适的问题上，人们表现得非常情绪化，实际上，就像鸡蛋的问题一样，没有技术上的原因来选择字节顺序规则，因此，争论沦为关于社会政治问题的争论，只要选择了一种规则并且始终如一地坚持，其实对于哪种字节排序的选择是任意的。

对于单一的字节（a byte），大部分处理器以相同的顺序处理位元（bit），因此单字节的存放方法和传输方式一般相同。

对于多字节数据，如整数（32位机中一般占4字节），在不同的处理器的存放方式主要有两种，以内存中0x0A0B0C0D的存放方式为例，分别有以下几种方式：

注：0x前缀代表十六进制。

大端序

大端序（英：big-endian）或称大尾序。

- 数据以8bit为单位：

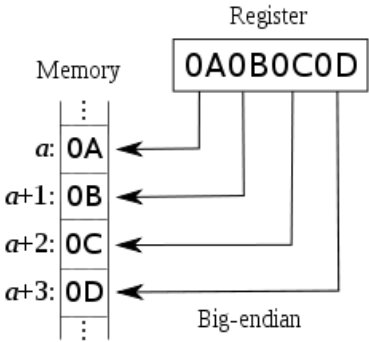
地址增长方向 →					
...	0x0A	0x0B	0x0C	0x0D	...

示例中，最高位字节是0x0A 存储在最低的内存地址处。下一个字节0x0B存在后面的地址处。正类似于十六进制字节从左到右的阅读顺序。

- 数据以16bit为单位：

地址增长方向 →			
...	0x0A0B	0x0C0D	...

最高的16bit单元0x0A0B存储在低位。



小端序

小端序（英：little-endian）或称小尾序。

- 数据以8bit为单位：

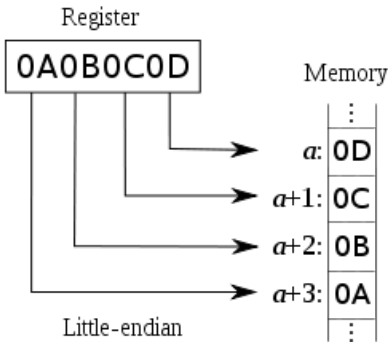
地址增长方向 →					
...	0x0D	0x0C	0x0B	0x0A	...

最低位字节是0x0D 存储在最低的内存地址处。后面字节依次存在后面的地址处。

- 数据以16bit为单位：

地址增长方向 →			
...	0x0C0D	0x0A0B	...

最低的16bit单元0x0D0C存储在低位。



- 更改地址的增长方向：

当更改地址的增长方向，使之由右至左时，表格更具有可阅读性。

← 地址增长方向					
...	0x0A	0x0B	0x0C	0x0D	...

最低有效位（LSB）是0x0D 存储在最低的内存地址处。后面字节依次存在后面的地址处。

← 地址增长方向			
...	0x0A0B	0x0C0D	...

最低的16bit单元0x0C0D存储在低位。

混合序

混合序（英：middle-endian）具有更复杂的顺序。以PDP-11为例，0x0A0B0C0D被存储为：

- 32bit在PDP-11的存储方式

地址增长方向 →					
...	0x0B	0x0A	0x0D	0x0C	...

可以看作最高的16bit位和低位以大端序存储，但16bit内部以小端存储。

处理器体系

- x86, MOS Technology 6502, Z80, VAX, PDP-11等处理器为Little endian。
- Motorola 6800, Motorola 68000, PowerPC 970, System/370, SPARC（除V9外）等处理器为Big endian
- ARM, PowerPC（除PowerPC 970外), DEC Alpha, SPARC V9, MIPS, PA-RISC and IA64的字节序是可配置的。

网络序

网络传输一般采用大端序，也被称之为网络字节序，或网络序。IP协议中定义大端序为网络字节序。

伯克利socket API定义了一组转换函数，用于16和32bit整数在网络序和本机字节序之间的转换。htonl，htons用于本机序转换到网络序；ntohl，ntohs用于网络序转换到本机序。

位序

一般用于描述串行设备的传输顺序。网络协议中只有数据链路层的底端会涉及到。

小端序（先传低位）的串行协议

- RS-232
- RS-422
- RS-485
- USB
- 以太网（虽然高字节先传，但每一字节内低位先传）

大端序（先传高位）的串行协议

- I2C协议
- SPI协议
- 摩尔斯电码

参见

- 最高有效位
- 最低有效位
- 网络协议

外部链接

- Endian的由来 (http://www.eygle.com/digest/2007/01/whats_mean_endian.html)
- White Paper: Endianness or Where is Byte 0? (<http://3bc.bertrand-blanc.com/endianness05.pdf>) （英文）
- Byte Ordering PPC (<http://developer.apple.com/documentation/CoreFoundation/Conceptual/CFMemoryMgmt/Concepts/ByteOrdering.html>) （英文）
- The Layout of Data in Memory (<http://www.steveheller.com/cppad/Output/basics7.html>) （英文）
- Writing endian-independent code in C (<http://www.ibm.com/developerworks/aix/library/au-endianc/index.html?ca=drs->) （英文）
- How to convert an integer to little endian or big endian (<http://techforb.blogspot.com/2007/10/how-to-convert-integer-to-little-endian.html>) （英文）
- Understanding big and little endian byte order (<http://betterexplained.com/articles/understanding-big-and-little-endian-byte-order/>) （英文）
- Mandatory reading: ON HOLY WARS AND A PLEA FOR PEACE (<http://www.ietf.org/rfc/ien/ien137.txt>) （英文）

参考

- ↑ 《深入理解计算机系统》 第2章 信息的表示和处理 P26.
- ↑ 《格列夫游记》 第一卷第4章 蒋剑锋（译）.
- ↑ 《深入理解计算机系统》 第2章 信息的表示和处理 P27.

取自“<http://zh.wikipedia.org/w/index.php?title=字节序&oldid=29340060>”

-
- 本页面最后修订于2013年11月27日（星期三）17:26。
 - 本站的全部文字在知识共享 署名-相同方式共享 3.0协议之条款下提供，附加条款亦可能应用（请参阅使用条款）。Wikipedia®和维基百科标志是维基媒体基金会的注册商标；维基™是维基媒体基金会的商标。维基媒体基金会是在美国佛罗里达州登记的501(c)(3)免税、非营利、慈善机构。