Endianness

In computing, endianness is the ordering of individually addressable sub-units (words, bytes, or even bits) within a longer data word stored in external memory. The most typical cases are the ordering of bytes within a 16-, 32-, or 64-bit word, where endianness is often simply referred to as byte order. The usual contrast is between the most versus the least significant byte first, called big-endian and little-endian respectively. Mixed forms are also possible; the ordering of bytes within a 16-bit word may be different from the ordering of 16-bit words within a 32-bit word, for instance; although rare, such cases are sometimes collectively referred to as mixed-endian or middle-endian.

Endianness may be seen as a low-level attribute of a particular representation format, for example, the order in which the two bytes of an UCS-2 character are stored in memory. Byte order is an important consideration in network programming, since two computers with different byte orders may be communicating. Failure to account for varying endianness when writing code for mixed platforms can lead to bugs that can be difficult to detect.

Source: https://en.wikipedia.org/wiki/Endianness

Big-endian and little-endian architecture

The adjectives big-endian and little-endian refer to which bytes are most significant in multi-byte data types and describe the order in which a sequence of bytes is stored in a computer’s memory.

In a big-endian system, the most significant value in the sequence is stored at the lowest storage address (i.e., first). In a little-endian system, the least significant value in the sequence is stored first. For example, consider the number 1025 (2 to the tenth power plus one) stored in a 4-byte integer:

00000000 00000000 00000100 00000001

Address	Big-Endian representation of 1025	Little-Endian representation of 1025
00	00000000	00000001
01	00000000	00000100
02	00000100	00000000
03	00000001	00000000

Many mainframe computers, particularly IBM mainframes, use a big-endian architecture. Most modern computers, including PCs, use the little-endian system. The PowerPC system is bi-endian because it can understand both systems.

Converting data between the two systems is sometimes referred to as the NUXI problem. Imagine the word UNIX stored in two 2-byte words. In a Big-Endian systems, it would be stored as UNIX. In a little-endian system, it would be stored as NUXI.

Note that the example above shows only big- and little-endian byte orders. The bit ordering within each byte can also be big- or little-endian, and some architectures actually use big-endian ordering for bits and little-endian ordering for bytes, or vice versa.

The terms big-endian and little-endian are derived from the Lilliputians of Gulliver's Travels, whose major political issue was whether soft-boiled eggs should be opened on the big side or the little side. Likewise, the big-/little-endian computer debate has much more to do with political issues than technological merits.

Source: https://www.webopedia.com