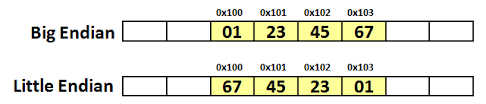
Big Endian and Little Endian

Big Endian and Litte Endian are two methods of storing multi-byte data types like int,float,etc.

In Little Endian last byte of binary representation are stored first of the word and in Big Endian first byte of binary represenation is stored as first byte of the word.

Suppose an integer is stored as 4 bytes(32-bits), then a variable y with value 0x01234567(Hexa-decimal representation)is stored as four bytes 0x01, 0x23, 0x45, 0x67, on Big-endian while on Little-Endian (Intel x86), it will be stored in reverse order as 0x67, 0x45, 0x23, 0x01.



Advantages and Disadvantages

In “Little Endian” form, assembly language instructions for picking up a 1, 2, 4, or longer byte number proceed in exactly the same way for all formats: first pick up the lowest order byte at offset 0. Also, because of the 1:1 relationship between address offset and byte number (offset 0 is byte 0), multiple precision math routines are correspondingly easy to write It is easier to add and multiply.

In “Big Endian” form, by having the high-order byte come first, you can always test whether the number is positive or negative by looking at the byte at offset zero. You don’t have to know how long the number is, nor do you have to skip over any bytes to find the byte containing the sign information. The numbers are also stored in the order in which they are printed out, so binary to decimal routines are particularly efficient.It is easy to compare numbers and also divide numbers.