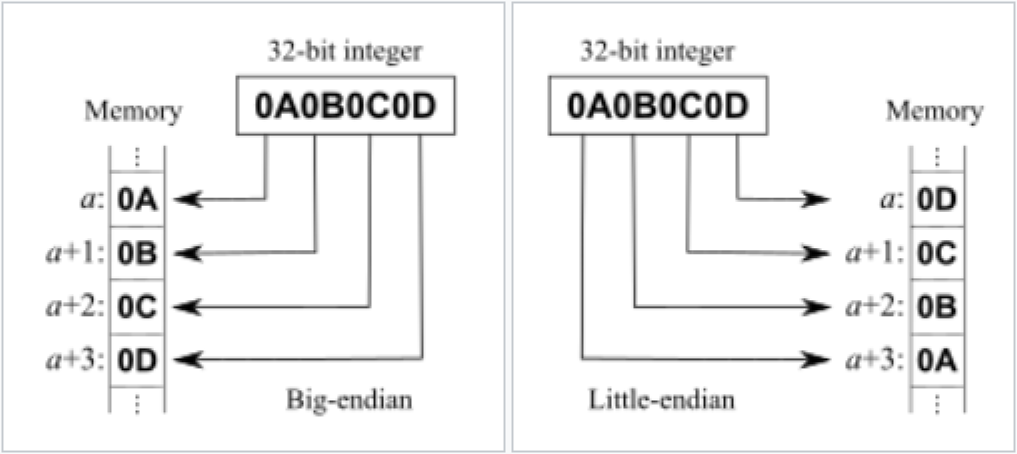
CS Studies - Endianness

> *“Computers store information in various-sized groups of binary bits. Each group is assigned a number, called its address. ...the smallest data group with an address is eight bits long and is called byte. Larger groups comprise two or more bytes. There are **two possible ways** a computer could **number the individual bytes in a larger group**, starting at either end. **Both types of endianness** are in widespread use in digital electronic engineering. The initial choice of endianness of a new design is **often arbitrary**, but later technology revisions and updates perpetuate the existing endianness to maintain **backward compatibility**.”* https://en.wikipedia.org/wiki/Endianness

As stated in Wikipedia, usually any computer can work equally well regardless of which endianness it uses since it will be consistent. However this can be an issue when data is moved externally between different computers. The endianness of the data must be understood and accounted for.

Endian example



Big-endian

Little-endian

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

Above diagram shows how two different computers can be using different endianness to store a 32-bit integer. Integer is broken into four bytes and bytes are stored in four sequential byte locations in memory. Big-endian stores the address with "0A" as the first memory location and little endian stores its last "0D" as its first memory location.

Usually, big-endianness is the dominant ordering in networking protocols, and little endianness is the dominant ordering for processor architectures (x86, most ARM implementations, base RISC-V implementations) and their associated memory.