

Endianess

Davide Gaetano

Novembre 2025

1 Introduzione

Nei processori **Intel**, i dati sono immagazzinati nella memoria RAM in modo invertito, ovvero in **little endian**. Vediamo un esempio:

- L'indirizzo **0x12345678** nella RAM sarà immagazzinato come **0x78 0x56 0x34 0x12**.
- Contrariamente, in **big endian** (ad esempio nel traffico di rete o nella maggior parte dei sistemi RISC) saranno memorizzati in questo modo: **0x12 0x34 0x56 0x78** (più leggibile dall'occhio umano).

Ovviamente bisogna ricordare che l'endianess vale solamente per la **memoria** (e non per i registri) e per i **byte** (non i bit).

Ecco alcune immagini per comprendere meglio:

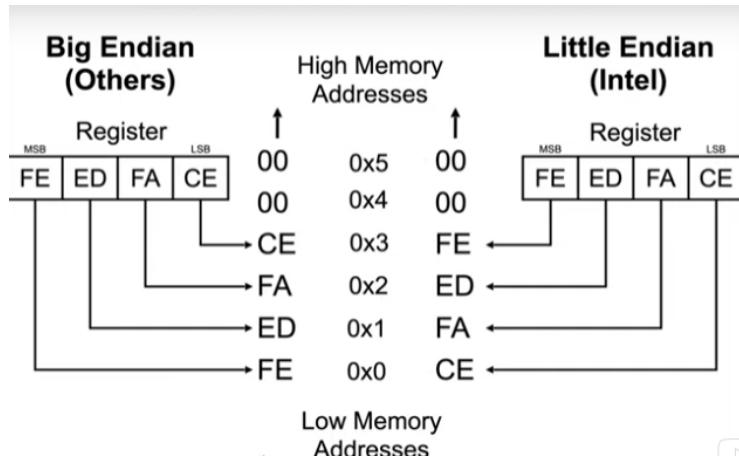
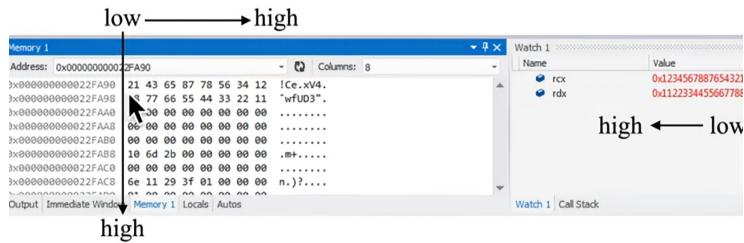


Figure 1: Spiegazione

How you'll probably usually see endianness expressed:



Memory dump windows are typically shown in typical English-writing left-to-right, top-to-bottom form, with the upper left being the lowest address

Register view windows always show the registers in big endian order

Figure 2: Output di un debugger (WinDbg) a singoli byte

Nella Figura 2 vediamo un esempio in un debugger (**WinDbg**) e notiamo i registri e i loro rispettivi valori disposti in memoria in **little endian** a singoli byte.

Attenzione: se chiediamo al nostro debugger di disporre questi valori in maniera diversa, l'output potrebbe variare.

Ecco un esempio di debugger che mostra i valori a blocchi di 4 byte:

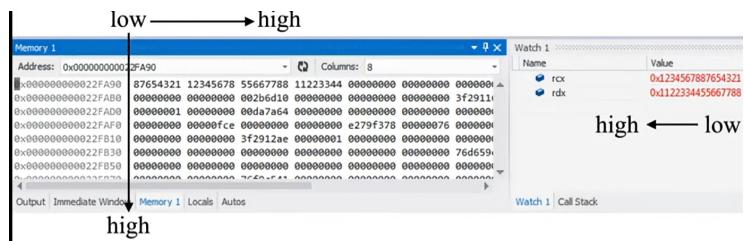


Figure 3: Output del debugger a blocchi di 4 byte

Ovviamente può essere facile confondersi poiché tutti i singoli blocchi di 4 byte sono in **little endian**.

Ecco un altro esempio a 8 byte alla volta:

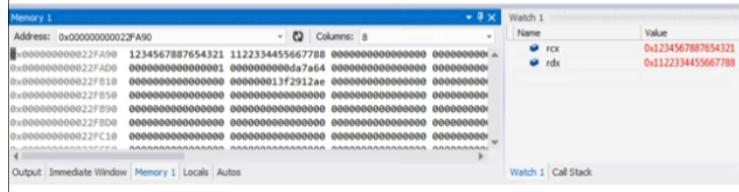


Figure 4: Output del debugger a blocchi di 8 byte

In questo caso è molto più leggibile, ma ricordiamoci sempre che parliamo di **little endian** (e quindi il “big endian” che vediamo è solo una rappresentazione del debugger per rendere l’output più leggibile).