**비트(bit)와 바이트(byte)**

컴퓨터는 모든 데이터를 2진수로 표현하고 처리

비트(bit)란 컴퓨터가 데이터를 처리하기 위해 사용하는 데이터의 최소 단위.

이러한 비트에는 2진수의 값(0과 1)을 단 하나만 저장

바이트(byte)란 위와 같은 비트가 8개 모여서 구성되며, 한 문자를 표현할 수 있는 최소

**바이트 저장 순서(byte order)**

컴퓨터는 데이터를 메모리에 저장할 때 바이트(byte) 단위로 나눠서 저장

하지만 컴퓨터가 저장하는 데이터는 대게 32비트(4바이트)나 64비트(8바이트)로 구성

따라서 이렇게 연속되는 바이트를 순서대로 저장해야 하는데, 이것을 바이트 저장 순서(byte order)

이때 바이트가 저장되는 순서에 따라 다음과 같이 두 가지 방식으로 나눔

1. 빅 엔디안(big endian)

2. 리틀 엔디안(little endian)

**빅 엔디안(big endian)**

빅 엔디안 방식은 낮은 주소에 데이터의 높은 바이트(MSB, Most Significant Bit)부터 저장하는 방식

이 방식은 평소 우리가 숫자를 사용하는 선형 방식과 같은 방식

따라서 메모리에 저장된 순서 그대로 읽을 수 있으며, 이해하기가 쉽다는 장점

SPARC을 포함한 대부분의 RISC CPU 계열에서는 이 방식으로 데이터를 저장

 Ex) 0x12345678

Output = 0x12, 0x34, 0x56, 0x78

**리틀 엔디안(little endian)**

리틀 엔디안 방식은 낮은 주소에 데이터의 낮은 바이트(LSB, Least Significant Bit)부터 저장하는방식

이 방식은 평소 우리가 숫자를 사용하는 선형 방식과는 반대로 거꾸로 읽음

대부분의 인텔 CPU 계열에서는 이 방식으로 데이터를 저장

Ex) 0x12345678

Output = 0x78, 0x56,0x34 ,0x12

**빅 엔디안 vs 리틀 엔디안**

빅 엔디안과 리틀 엔디안은 단지 저장해야 할 큰 데이터를 어떻게 나누어 저장하는가에 따른 차이

물리적으로 데이터를 조작하거나 산술 연산을 수행할 때에는 리틀 엔디안 방식이 더 효율적

하지만 데이터의 각 바이트를 배열처럼 취급할 때에는 빅 엔디안 방식이 더 적합합

네트워크를 통해 데이터를 전송할 때에는 빅 엔디안 방식이 사용

인텔 기반의 시스템에서 소켓 통신을 할 때는 바이트 순서에 신경을 써서 데이터를 전달