

## | D. REPRESENTACIÓN INTERNA Y ORDENAMIENTO DE BYTES: ENDIANNESS

| P1: En el área de inspección ¿Qué muestra el campo Int8 cuando se selecciona el primer byte del archivo? Y si selecciona los dos primeros bytes, ¿Qué se visualiza en Int16? ¿Cómo se explica esto?

Al seleccionar el primer byte del archivo (0x0A), el campo Int8 muestra el valor decimal 10, tanto en su representación con signo (signed) como sin signo (unsigned). Esto se debe a que 0x0A equivale al binario 0000 1010, cuyo cálculo es  $1 \times 2^3 + 1 \times 2^1 = 10$ . Al extender la selección a los primeros dos bytes (0A 00), el campo Int16 mantiene el valor 10. Bajo la arquitectura Little Endian, el primer byte es el menos significativo (LSB), por lo que la secuencia se interpreta como el valor hexadecimal 0x000A, manteniendo la misma equivalencia decimal.

| P2. Seleccione el cuarto y quinto byte en el área de edición. ¿Qué valor aparece en la interpretación como entero de 16 bits (Int16)? Luego, seleccione los bytes desde el cuarto hasta el séptimo. ¿Qué valor aparece ahora en la interpretación como entero de 32 bits (Int32)? Convierta manualmente cada uno de estos valores a su representación hexadecimal y explique por qué se muestran esos resultados.

| Cuarto y quinto bytes - 0x0138

Al situarnos en el cuarto byte (0x0138), el valor que aparece en la interpretación como entero de 16 bits es el valor decimal 312. Vamos a confirmar que este valor es correcto pasándolo de decimal a hexadecimal.

1. Conversión de decimal a hexadecimal

$$\frac{312}{16} = 19, \text{ residuo } 8$$

$$\frac{19}{16} = 1, \text{ residuo } 3$$

$$\frac{1}{16} = 0, \text{ residuo } 1$$

Resultado hexadecimal: 0x138

Sabiendo que queremos ver el valor del entero en 16 bits, agregamos un cero a la izquierda para completar los 2 bytes (actualmente tenemos 3 dígitos hexadecimales, es decir, 12 bits).

Resultado hexadecimal: 0x01 38

| Cuarto al séptimo byte - 0x00000138

Cuando seleccionamos desde el cuarto byte (0x38) hasta el séptimo (0x00), observamos en la interpretación de 32 bits el valor decimal 312. Como se puede apreciar, es el mismo valor que en la respuesta anterior. Esto se debe a que, al utilizar el formato Little Endian, simplemente estamos agregando dos bytes adicionales (cuatro dígitos hexadecimales en cero) como parte más significativa, por lo que el valor decimal resultante es el mismo. Vamos a convertir el valor decimal considerando los 32 bits para comprobarlo.

1. Conversión de decimal a hexadecimal

$$\frac{312}{16} = 19, \text{ residuo } 8$$

$$\frac{19}{16} = 1, \text{ residuo } 3$$

$$\frac{1}{16} = 0, \text{ residuo } 1$$

Resultado hexadecimal: 0x138

Sabiendo que queremos representar el valor como un entero de 32 bits, agregamos cinco ceros a la izquierda para completar los 4 bytes (es decir, un total de 8 dígitos hexadecimales).

Resultado hexadecimal: 0x00 00 01 38

De este modo, obtenemos el mismo valor hexadecimal que muestra el programa.

Aquí tienes el texto corregido, con la ortografía, acentuación y redacción optimizadas, además de ajustar el formato de las expresiones matemáticas:

## | P3. Seleccione los últimos cuatro bytes y observe el valor que aparece en la interpretación como entero de 32 bits (Int32). ¿Por qué se muestra ese valor? Realice las conversiones manualmente para comprender el motivo del resultado.

Los últimos cuatro bytes corresponden en el programa a `0xC8 FE FF FF`. Sabiendo que se utiliza el formato **Little Endian**, el valor hexadecimal original (reordenado) sería `0xFF FF FE C8`. La interpretación como entero de 32 bits para estos bytes es 4 294 966 984 en formato sin signo (*unsigned*) y -312 en formato con signo (*signed*). Vamos a convertir cada uno de estos valores decimales de vuelta a hexadecimal para comprobarlo.

### | Unsigned - 4 294 966 984

Vamos a convertir el valor decimal 4 294 966 984 directamente a hexadecimal:

$$\frac{4294966984}{16} = 268435436, \text{ residuo } 8$$

$$\frac{268435436}{16} = 16777214, \text{ residuo } 12 (\text{C})$$

$$\frac{16777214}{16} = 1048575, \text{ residuo } 14 (\text{E})$$

$$\frac{1048575}{16} = 65535, \text{ residuo } 15 (\text{F})$$

$$\frac{65535}{16} = 4095, \text{ residuo } 15 (\text{F})$$

$$\frac{4095}{16} = 255, \text{ residuo } 15 (\text{F})$$

$$\frac{255}{16} = 15, \text{ residuo } 15 (\text{F})$$

$$\frac{15}{16} = 0, \text{ residuo } 15 (\text{F})$$

Resultado hexadecimal final: 0xFF FF FE C8

Como podemos observar, el resultado es correcto.

### | Signed (-312)

Para convertir -312 a hexadecimal, primero pasaremos el valor a binario usando el sistema de **complemento a dos** y luego realizaremos la conversión a hexadecimal.

#### | 1. Convertir 312 (positivo) a binario de 32 bits

$$312_{10} = 0000\ 0000\ 0000\ 0000\ 0001\ 0011\ 1000_2$$

#### | 2. Invertir los bits (NOT)

$$1111\ 1111\ 1111\ 1111\ 1111\ 1110\ 1100\ 0111_2$$

### I 3. Sumar 1 (Complemento a dos)

1111 1111 1111 1111 1111 1110 1100 1000<sub>2</sub>

Finalmente, agrupamos los bits de cuatro en cuatro para obtener el valor hexadecimal:

- 1111 = F
- 1111 = F
- 1111 = F
- 1111 = F
- 1110 = E
- 1100 = C
- 1000 = 8

**Resultado hexadecimal final:** 0xFF FF FE C8.

Nuevamente, confirmamos que el resultado coincide con el valor interpretado por el programa.