PROGRAMACIÓN AVANZADA

Operadores bit a bit

CONTENIDO

Manipulación de datos usando operadores bit a bit.

¿POR QUÉ MANIPULAR DATOS A NIVEL DE BIT?

Los datos en la computadora están representados en formato binario.

¿Cuándo debo usar una codificación binaria personalizada?

- Si desea guardar información de una manera más compacta.
- Si desea guardar información en un formato que sea independiente de cualquier tecnología digital.
- Si desea agregar seguridad personalizada mediante un formato de datos propietario.

PATRÓN DE BITS DE UN TIPO ENTERO

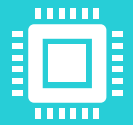
The bits position is numerated from right to left starting with 0 (least significant bit).
Example:

- The character * is encoded to the value 42 in ASCII code and in binary it is represented as:

Bits pattern	0 0 1 0 1 0 1 0
Bits position	7 6 5 4 3 2 1 0

MANIPULACIÓN DE DATOS A NIVEL DE BITS

La manipulación de datos a nivel de bit a menudo se realiza utilizando enteros sin signo.



La manipulación de datos a nivel de bit depende del hardware.

OPERADORES BIT A BIT

Operador	Significado	Ejemplo	Resultado
&	Conjunction (AND)	x & y	Los bits en el resultado se establecen en 1 si los bits correspondientes en x e y son ambos 1.
|	Disjunction (OR)	x | y	Los bits en el resultado se establecen en 1 si al menos uno de los bits correspondientes en x e y es 1.
^	Exclusive disjunction (XOR)	x ^ y	Los bits en el resultado se establecen en 1 si uno de los bits correspondientes en x e y es 1 y el otro bit es 0.
~	One's complement	~x	Todos los 0 bits en x se establecen en 1 y todos los 1 bits se establecen en 0.
<<	Left shift	x << n	Desplaza los bits del primer operando a la izquierda por el número de bits especificado por el segundo operando. Se llena desde la derecha con 0 bits.
>>	Right shift	x >> n	Desplaza los bits del primer operando a la derecha por la cantidad de bits especificados por el segundo operando. El método de llenado desde la izquierda depende de la máquina cuando el operando izquierdo es negativo.

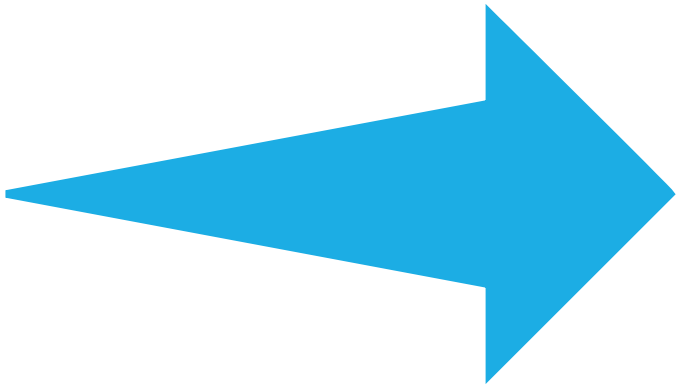
Expresion (o declaracion)	Bits pattern
`unsigned int a = 6;`	0 … 0 0 1 1 0
`unsigned int b = 11;`	0 … 0 1 0 1 1
`a & b`	0 … 0 0 0 1 0
`a | b`	0 … 0 1 1 1 1
`a ^ b`	0 … 0 1 1 0 1
`~a`	1 … 1 1 0 0 1

EJEMPLOS

MÁS OPERADORES

- `x &= y;` `// Equivalente a x = x & y;`
- `x |= y;` `// Equivalente a x = x | y;`
- `x ^= y;` `// Equivalente a x = x ^ y;`
- `x <<= y;` `// Equivalente a x = x << y;`
- `x >>= y;` `// Equivalente a x = x >> y;`
- Ejemplo:
  - `unsigned int n = 0xB;` `// Bit pattern: 0… 0 0 0 1 0 1 1`
  - `n <<= 2;` `// 0… 0 1 0 1 1 0 0`
  - `n >>= 2;` `// 0… 0 0 0 0 0 1 0`

DE BINARIO A HEXADECIMAL



https://es.wikihow.com/convertir-un-binario-en-hexadecimal

APLICANDO MÁSCARAS

```
a &= ~0x20;        // Set the bit 5 to zero of a. 0x20
                      in binary is 10 0000
int mask = 0xC;    // 0xC is 1100
a |= mask;         // Set to 1 the bits 2 and 3 in a.
a ^= mask;         ¿Qué hace esto?
```

EJERCICIOS

1. Realice un programa en C que garantice que los tres bits de menor peso de un byte sean 0.

2. Realice un programa en C que compruebe el valor del bit 12 de una variable *var* de tipo *short in*.

3. Realice un programa en C para extraer el número binario de 3 bits contenido en los tres bits de menor peso de una variable.

MUCHAS GRACIAS