

OPERACIONES A NIVEL DE BIT

Recordemos que las computadoras almacenan los datos en sistema binario. Es decir almacenan usando 0 y 1.

Nosotros como programadores, podemos acceder a esta información a través de las operaciones a nivel de bit.

Ahora aprenderemos todas ellas y sus usos.

1. Operaciones a Nivel de Bit

a) AND (&)

p	q	p & q
1	1	1
1	0	0
0	1	0
0	0	0

Es idéntico al Y (AND) lógico, pero este trabaja a nivel de bits.

Ej:

8	7	6	5	4	3	2	1	0		
1	0	1	1	1	0	1	0	0	372	
1	1	0	1	0	0	1	0	1	421	
1	0	0	1	0	0	1	0	0	372 & 421 =	292

Recordemos:
int tiene 32 bits

b) OR (|)

p	q	p q
1	1	1
1	0	1
0	1	1
0	0	0

Es idéntico al O (OR) lógico, pero este trabaja a nivel de bits.

Ej:

8	7	6	5	4	3	2	1	0		
1	0	1	1	1	0	1	0	0	372	
1	1	0	1	0	0	1	0	1	421	
1	1	1	1	1	0	1	0	1	372 421 =	501

c) XOR (^)

p	q	p ^ q
1	1	0
1	0	1
0	1	1
0	0	0

Es llamado el "o exclusivo"

Ej:

8	7	6	5	4	3	2	1	0		
1	0	1	1	1	0	1	0	0	372	
1	1	0	1	0	0	1	0	1	421	
0	1	1	0	1	0	0	0	1	372 ^ 421 =	209

d) NOT (~)

P	~ P
1	0
0	1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	1	0	0		372
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	0	1	0	1	1	~ 372	-373

e) Desplazamiento hacia adelante (<<)

10 = 1 0 1 0

10 << 3 = 1 0 1 0 0 0 0 = 80 10 x 2³

f) Desplazamiento hacia atras (>>)

10 = 1 0 1 0

10 >> 3 = 1 10 / 2³

2. USOS

a) Saber el estado de un bit

```

      7 6 5 4 3 2 1 0
      1 0 1 1 1 1 0 0
      &
          1 0 0 0 0 0
      -----
      0 0 0 1 0 0 0 0
  
```

El bit 4 esta prendido

```

      7 6 5 4 3 2 1 0
      1 0 1 1 1 1 0 0
      &
          1 0 0 0 0 0 0 0
      -----
      0 0 0 0 0 0 0 0
  
```

El bit 6 esta apagado

$n \& (1 \ll pos)$

 ↗ 0 apagado

 ↘ != 0 prendido

Aplicación: Imprimir todas los bits de una variable

31 porque int tiene 32 bits

```
// Imprimamos todos los bits de una variable
for(int i=31;i>=0;i--){
    if((x&(1<<i)) == 0){
        cout<<0;
    }
    else{
        cout<<1;
    }
}
```

b) Prender un bit

pos

	5	4	3	2	1	0
	1	0	0	1	1	0
OR	1		1	0	0	0
	1	0	1	1	1	0

pos

$n | (1 \ll pos)$

	5	4	3	2	1	0
n =	1	0	0	1	1	0
				1	0	0
	1	0	1	1	1	0

c) Apagar un bit

pos

	5	4	3	2	1	0
	1	1	0	0	1	1
AND	1	0	1	1	1	1
	1	0	0	0	1	1

$n \& (\sim (1 \ll pos))$

pos

	5	4	3	2	1	0
	1	1	0	0	1	1
AND	1	1	0	1	1	1
	1	1	0	0	1	1

0 → 1 1 → 0

TAREA EXTRA (5 pts)

¿Cuál sería la fórmula para switchear un bit en cierta posición, es decir si es 0 se vuelve 1 y si es 1 se vuelve 0?