

# Programação em C

Operadores bit a bit

**Agostinho Brito**

2022

- Motivação: imagine uma aplicação onde se deseja controlar uma faixa de LEDs através usando um microcontrolador programável em C.



- Para controlar os LEDs poderíamos usar de um array de chars, onde cada posição do array representa um LED e o valor guardado na posição indica se o LED está aceso ou apagado.

```
char leds[8] = {1, 0, 0, 1, 1, 0, 1, 0};
```

- Nesse exemplo, os LEDs 0, 3, 4 e 6 estariam acesos.



- Perceba que cada elemento do array `leds[]` guarda apenas dois tipos de **INFORMAÇÃO** :  
0 ou 1. Logo, poderíamos representar cada elemento do array com apenas um bit.
- Entretanto, o C não possui um tipo de dado que representa um bit.
- Solução: usar operadores bit-a-bit para codificar valores booleanos em um tipo de dado que representa um byte.

## O que são operadores

- Operadores são símbolos que informam ao compilador que operação lógica ou matemática se deseja realizar com um ou mais operandos.
- Dos vários operadores da linguagem C, os operadores bit-a-bit são os que operam diretamente sobre os bits de um dado, sendo capazes de acessá-los ou modificá-los de forma independente.
- Os operadores bit a bit são: `&`, `|`, `^`, `<<`, `>>`, `~`, `^=`, `<<=`, `>>=`.
- Operar diretamente sobre os bits de um dado é muito útil para se obter um desempenho melhor em algumas aplicações, principalmente em aplicações que envolvam *hardware* ou *software* de baixo nível.
- Em operações de comunicação de dados pela Internet, por exemplo, operadores bit-a-bit são usados a todo instante para codificar informações.

# Apresentando as operações bit-a-bit

- O operador `&` é o operador de AND bit-a-bit.
- O operador `|` é o operador de OR bit-a-bit.
- O operador `^` é o operador de XOR bit-a-bit.
- O operador `<<` é o operador de deslocamento para a esquerda.
- O operador `>>` é o operador de deslocamento para a direita.
- O operador `~` é o operador de complemento de um.
- O operador `^=` é o operador de XOR bit-a-bit com atribuição.
- O operador `<<=` é o operador de deslocamento para a esquerda com atribuição.
- O operador `>>=` é o operador de deslocamento para a direita com atribuição.

- Considere os números decimais 186 e 238, representados em duas variáveis inteiras não sinalizadas do tipo char.

```
unsigned char x=186, y=238;
```

- Seus equivalentes binários são os números **10111010** (186) e **11101110** (238).
- As operações bit-a-bit são realizadas sobre os bits de cada número, e não sobre os números em si. As operações realizadas com os símbolos `&`, `|`, `^` resultam valores conforme a tabela seguinte:

bit	7	6	5	4	3	2	1	0
x	1	0	1	1	1	0	1	0
y	1	1	1	0	1	1	1	0
<b>x&amp;y</b>	1	0	1	0	1	0	1	0
<b>x y</b>	1	1	1	0	1	1	1	0
<b>x^y</b>	0	1	0	1	0	1	0	0

Operador	Descrição	Exemplo
&	AND binário	A & B
	OR binário	A   B
^	XOR binário	A ^ B
~	Complemento de 1	~A
<<	Deslocamento à esquerda	A << B
>>	Deslocamento à direita	A >> B

- Assumindo, por exemplo, que duas variáveis `unsigned char` tenham valores numéricos  $A = 115$  e  $B = 37$ , como ficam os resultados dos operadores bit-a-bit?

Operador	Descrição	Exemplo
&	AND binário	A & B
	OR binário	A   B
^	XOR binário	A ^ B
~	Complemento de 1	~A
<<	Deslocamento à esquerda	A << B
>>	Deslocamento à direita	A >> B

- Assumindo, por exemplo, que duas variáveis `unsigned char` tenham valores numéricos  $A = 115$  e  $B = 37$ , como ficam os resultados dos operadores bit-a-bit?



## Praticando o uso de operadores...



Obrigado