

Operadores Relacionais na Linguagem C

1. Introdução

Os operadores relacionais em C são usados para comparar dois valores. Eles retornam um valor booleano (1 para verdadeiro e 0 para falso) dependendo da relação entre os operandos. São fundamentais para estruturas de controle como `if`, `while` e `for`.

2. Lista de Operadores Relacionais

Operador	Descrição	Exemplo (a = 10, b = 20)	Resultado
<code>==</code>	Igual a	<code>a == b</code>	0 (falso)
<code>!=</code>	Diferente de	<code>a != b</code>	1 (verdadeiro)
<code>></code>	Maior que	<code>a > b</code>	0 (falso)
<code><</code>	Menor que	<code>a < b</code>	1 (verdadeiro)
<code>>=</code>	Maior ou igual a	<code>a >= b</code>	0 (falso)
<code><=</code>	Menor ou igual a	<code>a <= b</code>	1 (verdadeiro)

3. Uso Prático dos Operadores Relacionais

Os operadores relacionais são comumente usados em expressões condicionais.

Exemplo 1: Uso com `if`

```
#include <stdio.h>

int main() {
    int idade = 18;

    if (idade >= 18) {
        printf("Você é maior de idade.\n");
    } else {
        printf("Você é menor de idade.\n");
    }

    return 0;
}
```

Saída:

```
Você é maior de idade.
```

Exemplo 2: Uso em um loop **while**

```
#include <stdio.h>

int main() {
    int contador = 0;

    while (contador < 5) {
        printf("Contador: %d\n", contador);
        contador++;
    }

    return 0;
}
```

Saída:

```
Contador: 0
Contador: 1
Contador: 2
Contador: 3
Contador: 4
```

4. Cuidados ao Usar Operadores Relacionais

4.1. Diferença entre **=** e **==**

Um erro comum é confundir **=** (atribuição) com **==** (comparação). Veja o exemplo incorreto:

```
if (x = 5) { // Erro: x recebe 5, e a condição sempre será verdadeira
```

O correto seria:

```
if (x == 5) { // Correta comparação
```

4.2. Comparação com números de ponto flutuante

Devido à imprecisão dos números de ponto flutuante, comparações com **==** podem não funcionar corretamente.

```
float a = 0.1 + 0.2;
if (a == 0.3) { // Pode falhar devido a erros de precisão
    printf("Iguais\n");
}
```

A solução é usar uma margem de erro:

```
if (fabs(a - 0.3) < 0.0001) {
```

5. Conclusão

Os operadores relacionais são essenciais para a tomada de decisões em C. É importante usá-los corretamente, evitando erros comuns como a confusão entre `=` e `==` e problemas de precisão com números de ponto flutuante. Saber utilizá-los de forma eficiente melhora a lógica e a robustez do programa.