

# Estrutura de Controle

## 1. Condicionais

### Definição de Condicionais

As **estruturas condicionais**, também conhecidas como **estruturas de seleção**, são componentes essenciais na programação, pois permitem que o programa tome decisões com base em condições específicas. Elas avaliam expressões booleanas, ou seja, condições que podem ser verdadeiras ou falsas, e, dependendo do resultado dessa avaliação, executam diferentes blocos de código. As estruturas condicionais são fundamentais para tornar os programas dinâmicos e adaptáveis a diferentes situações.

Em C, os principais tipos de estruturas condicionais são o **if**, o **else** e o **else if**. O **if** é utilizado para verificar se uma condição é verdadeira. Caso seja, o bloco de código associado a ele será executado. Caso contrário, a execução do código continua após o **if**. Já o **else** serve como uma alternativa ao **if**, executando um bloco de código quando a condição do **if** não é atendida. O **else if** permite verificar múltiplas condições em sequência, testando uma nova condição se a anterior for falsa, e assim por diante. Caso nenhuma das condições seja atendida, o bloco de código associado ao **else** será executado, caso este esteja presente.

Essas estruturas tornam possível o controle do fluxo de execução de um programa, possibilitando que ele altere seu comportamento com base em diferentes entradas ou situações. O uso de condicionais é essencial para a criação de programas que precisam responder de maneira diferente a diferentes dados ou condições, sendo, portanto, uma das principais ferramentas para introduzir a lógica de tomada de decisão no desenvolvimento de software.

### Tipos de Condicionais

1. **if**: Executa um bloco de código se a condição for verdadeira.

```
#include <stdio.h>
```

```
int main() {  
    int idade = 20;  
  
    if (idade >= 18) {  
        printf("Você é maior de idade.\n");  
    }
```

```
    }

    return 0;
}
```

Neste exemplo, se a variável `idade` for maior ou igual a 18, a mensagem "Você é maior de idade." é exibida.

2. **`if...else`**: Executa um bloco de código se a condição for verdadeira, e outro bloco se a condição for falsa.

```
#include <stdio.h>

int main() {
    int nota = 75;

    if (nota >= 60) {
        printf("Você foi aprovado!\n");
    } else {
        printf("Você foi reprovado!\n");
    }

    return 0;
}
```

Nesse exemplo, se a nota é maior ou igual a 60, o estudante é aprovado. Caso contrário, é reprovado.

3. **`if...else if`**: Permite testar várias condições, executando diferentes blocos de instrução conforme necessário.

c

Copiar código

```
#include <stdio.h>

int main() {
    int hora = 14;

    if (hora < 12) {
        printf("Bom dia!\n");
    } else if (hora < 18) {
```

```
        printf("Boa tarde!\n");
    } else {
        printf("Boa noite!\n");
    }

    return 0;
}
```

Neste exemplo, dependendo do valor de `hora`, a mensagem de saudação é exibida.

---

## 2. Laços de Repetição

### Definição de Laços de Repetição

Os laços de repetição, também conhecidos como loops, permitem executar um bloco de código repetidamente enquanto uma determinada condição é verdadeira. Existem três principais tipos de laços de repetição em C:

1. **for**: Um laço controlado com um número fixo de iterações.

```
#include <stdio.h>

int main() {
    for (int i = 0; i < 5; i++) {
        printf("Iteração %d\n", i);
    }

    return 0;
}
```

Neste exemplo, a mensagem "Iteração 0" a "Iteração 4" será exibida cinco vezes.

2. **while**: Executa um bloco de código enquanto a condição especificada for verdadeira.

```
#include <stdio.h>

int main() {
    int i = 0;
```

```
while (i < 5) {
    printf("Valor de i: %d\n", i);
    i++;
}

return 0;
}
```

Aqui, o loop é executado enquanto **i** é menor que 5.

3. **do...while**: Similar ao **while**, mas garante que o bloco de código seja executado pelo menos uma vez.

```
#include <stdio.h>

int main() {
    int i = 0;

    do {
        printf("Valor de i: %d\n", i);
        i++;
    } while (i < 5);

    return 0;
}
```

Este exemplo também exibe "Valor de i" de 0 a 4, mas sempre pelo menos uma vez.

---

## 3. Exemplos Práticos

### Exemplo 1: Calculando a soma de números pares

```
#include <stdio.h>

int main() {
    int soma = 0;
```

```
for (int i = 1; i <= 10; i++) {  
    if (i % 2 == 0) { // Se i é par  
        soma += i;  
    }  
}  
  
printf("A soma dos números pares de 1 a 10 é: %d\n", soma);  
  
return 0;  
}
```

Este exemplo mostra o uso de um laço `for` para somar números pares de 1 a 10.

## Exemplo 2: Verificação de nota

```
#include <stdio.h>  
  
int main() {  
    int nota;  
  
    printf("Digite a nota do aluno: ");  
    scanf("%d", &nota);  
  
    if (nota >= 50) {  
        printf("Aluno aprovado!\n");  
    } else {  
        printf("Aluno reprovado!\n");  
    }  
  
    return 0;  
}
```

Neste exemplo, usamos uma estrutura de condição `if...else` para verificar se a nota do aluno é suficiente para ser aprovado.

---

## Referências

- KERNIGHAN, B. W.; RITCHIE, D. M. *The C Programming Language*. 2. ed. Englewood Cliffs: Prentice Hall, 1988.