

Alessandro Calin

Lista Encadeadas



ESTRUTURA DE DADOS
alessandrocalin@gmail.com
967617845

Lista encadeada é uma estrutura de dados frequentemente utilizada em programação e ciência da computação. É composta por uma sequência de elementos chamados nós, onde cada nó armazena dados e uma referência (ou ponteiro) para o próximo nó na sequência.

Existem dois tipos principais de listas encadeadas: simplesmente encadeadas e duplamente encadeadas.

- **Lista Encadeada Simples:** Neste tipo, cada nó possui apenas um ponteiro que aponta para o próximo nó na lista. A lista começa com um nó chamado "cabeça", que atua como o ponto de entrada.
- **Lista Encadeada Dupla:** Cada nó possui dois ponteiros: um que aponta para o próximo nó e outro que aponta para o nó anterior na sequência. Isso permite percorrer a lista em ambas as direções.

Vantagens:

- **Inserção e exclusão eficientes:** Adicionar ou remover elementos no início ou no meio da lista é rápido e requer apenas ajustes de ponteiros.
- **Flexibilidade:** As listas encadeadas podem crescer e diminuir dinamicamente durante a execução do programa.
- **Utilização eficiente de memória:** Elas alocam memória dinamicamente conforme necessário, ao contrário de algumas estruturas de dados estáticas.

Desvantagens:

- **Acesso aleatório lento:** Para acessar um elemento em uma lista encadeada, você normalmente precisa percorrer a lista a partir do início até o elemento desejado. Isso pode ser lento em comparação com estruturas de dados como arrays.
- **Uso de memória extra:** Cada nó em uma lista encadeada requer espaço adicional de memória para armazenar o ponteiro para o próximo nó.
- **Fragilidade:** A manipulação de ponteiros requer cuidado para evitar erros de programação, como referências nulas ou loops infinitos.

As listas encadeadas são frequentemente usadas em situações onde a inserção e a remoção de elementos são operações comuns, e onde o acesso aleatório não é tão importante quanto a flexibilidade e a eficiência no uso de memória.

- **Estruturas de Dados:** As listas encadeadas são frequentemente usadas como uma estrutura de dados fundamental para implementar outras estruturas mais complexas, como pilhas, filas, árvores e grafos.
- **Gerenciamento de Memória:** Em linguagens de programação de baixo nível, como C e C++, as listas encadeadas podem ser usadas para implementar alocação e desalocação dinâmica de memória, como em alocação de blocos de memória de tamanho variável.
- **Sistemas Operacionais:** Em sistemas operacionais, listas encadeadas podem ser usadas para gerenciar processos, threads, recursos e outras estruturas de dados importantes.
- **Bancos de Dados:** Em bancos de dados relacionais, as listas encadeadas são usadas para implementar índices, especialmente em sistemas de banco de dados que usam a estratégia de encadeamento para lidar com colisões em tabelas hash.

- **Edição de Texto:** Editores de texto e processadores de texto podem usar listas encadeadas para implementar funcionalidades como histórico de comandos, desfazer/refazer operações e listas de marcadores.
- **Jogos:** Em jogos de computador, as listas encadeadas podem ser usadas para representar várias estruturas de dados, como listas de personagens, listas de objetos em cena, e assim por diante.

Em C, a função `malloc()` é usada para alocar memória dinamicamente durante a execução do programa. Ela permite que você aloque uma quantidade específica de memória conforme necessário, em oposição à alocação estática de memória, que é feita em tempo de compilação.

Em C, `sizeof` é um operador que retorna o tamanho, em bytes, de um determinado tipo de dado ou de uma variável específica. Ele é frequentemente utilizado para determinar o tamanho de tipos de dados em tempo de compilação e para calcular o tamanho de arrays e estruturas de dados.



```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
typedef struct lista{
    int dado;
    struct lista *link;
}no;
```




```
no *crialista(int n){
    no *ini, *p, *ult;
    int valor;
    ini=ult=NULL;
    for(int i=1;i<=n;i++){
        printf("\nDigite o valor %d da lista ", i);
        scanf("%d",&valor);
        p=(no*) malloc (sizeof(no));
        p->dado=valor;
        p->link=NULL;
        if(ult!=NULL)
            ult->link=p;
        else
            ini=p;
        ult =p;
    }
    return ini;
}
```



```
void escrevelista(no *p){  
    while (p!=NULL){  
        printf("%d\t", p-> dado);  
        p=p->link;  
    }  
}
```



```
int main(){  
    no *primeira;  
    int n;  
    printf("Criando uma lista encadeada");  
    do{  
        printf("\nEntre com o numero de nois: ");  
        scanf("%d",&n);  
    }while(n<0);  
    primeira = crialista(n);  
    if(primeira != NULL){  
        printf("\nLista Criada");  
        printf("\n");  
        escrevelista(primeira);  
    }  
    else  
        printf("\nLista Vazia");  
    return(0);  
}
```

