



Revisão: Listas Estáticas e Dinâmicas em C

Esta aula abordará os conceitos de listas estáticas e dinâmicas em C. Exploraremos as diferenças, vantagens e desvantagens de cada tipo. Ao final, teremos duas atividades de implementação prática.

PW

por Prof. William



O que são Listas? (Conceito Geral)



Coleção de Dados

Listas são coleções de elementos organizados sequencialmente.



Relação Sequencial

Cada elemento está ligado ao próximo, formando uma ordem.



Tipos Variados

Podem ser estáticas ou dinâmicas, dependendo da alocação de memória.

Listas Estáticas: Definição e Características

Tamanho Fixo

O tamanho da lista é definido em tempo de compilação.

Alocação Contígua

Os elementos são armazenados em posições de memória adjacentes.

Simplicidade

Implementação mais simples em comparação com listas dinâmicas.

Vantagens e Desvantagens das Listas Estáticas

Vantagens

- Acesso rápido aos elementos.
- Implementação simples.
- Menor overhead de memória.

Desvantagens

- Tamanho limitado.
- Dificuldade em redimensionar.
- Desperdício de memória se subutilizada.

Implementação de Listas Estáticas em C (Exemplo Simples)

```
#include <stdio.h>
#define MAX_SIZE 10

int lista[MAX_SIZE];
int tamanho = 0;

int main() {
    // Adicionar elementos
    lista[tamanho++] = 5;
    printf("%d\n", lista[0]);
    return 0;
}
```

Listas Dinâmicas: Definição e Características

Tamanho Variável

O tamanho da lista pode ser alterado em tempo de execução.

Alocação Não Contígua

Elementos podem estar em posições de memória não adjacentes.

Flexibilidade

Maior flexibilidade no gerenciamento de dados.

Vantagens e Desvantagens das Listas Dinâmicas

Vantagens

- Tamanho flexível e adaptável.
- Utilização eficiente da memória.
- Facilidade em adicionar e remover elementos.

Desvantagens

- Implementação mais complexa.
- Maior overhead de memória.
- Acesso mais lento aos elementos.

Alocação Dinâmica de Memória (malloc, calloc, free)



1 malloc

Aloca um bloco de memória não inicializada.

2 calloc

Aloca um bloco de memória inicializada com zeros.

3 free

Libera a memória alocada dinamicamente.

Implementação de Listas Dinâmicas em C (Exemplo Simples)

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

struct Node {
    int data;
    struct Node* next;
};

int main() {
    struct Node* head = malloc(sizeof(struct Node));
    head->data = 10;
    head->next = NULL;
    printf("%d\n", head->data);
    free(head);
    return 0;
}
```

Dynamic Linked List

Last (ist ane lase) for this natch ast connued list, Ugltniext = (at) lists, the petingy.

Presceniay,
dynamic list (dynamic) selinked list),
ar linked list list).

Do the watl thæ, conamer to eutal of
dulling linket list, for of 1185/fictions.

...
List dynamied (ist(intenstreufaneed(idynamers
byr dethindyrrermer list list))
fit with dist,aetertions,(dynamiaallibla list)

Dynamic list list list;

(lssst liske it, buture = mning/(rallow and lisst on
there frught, a dynant list),-shut world),

Linkic List Linked list,
Celanciac list list ab; isr lite,
dynamic list wanted (prants on and rixe,
dynaëing grostrens.

Charse dynamic linked list)

Your friramuting and linker dynamic, dynamtric list, train)
anvubt.thong.cutldoring, Walst,declits and tsunlly in it.;

Static Lists	No memory allocation	Dynamic Lists
Performance	Memory is wasted in dynamical lists	Dynamic lists are efficient (no need for any allocations)
Flexibility	Memory is wasted in dynamical lists	Dynamic lists are efficient (no need for any allocations)

Comparação entre Listas Estáticas e Dinâmicas

Característica	Listas Estáticas	Listas Dinâmicas
Tamanho	Fixo	Variável
Alocação	Contígua	Não Contígua
Flexibilidade	Baixa	Alta
Complexidade	Simples	Complexa



Quando usar Listas Estáticas?

1 Tamanho Conhecido

Quando o número de elementos é conhecido e não varia.

2 Simplicidade

Quando a simplicidade da implementação é crucial.

3 Restrições de Memória

Em sistemas com restrições de memória.

Quando usar Listas Dinâmicas?

1 Tamanho Desconhecido

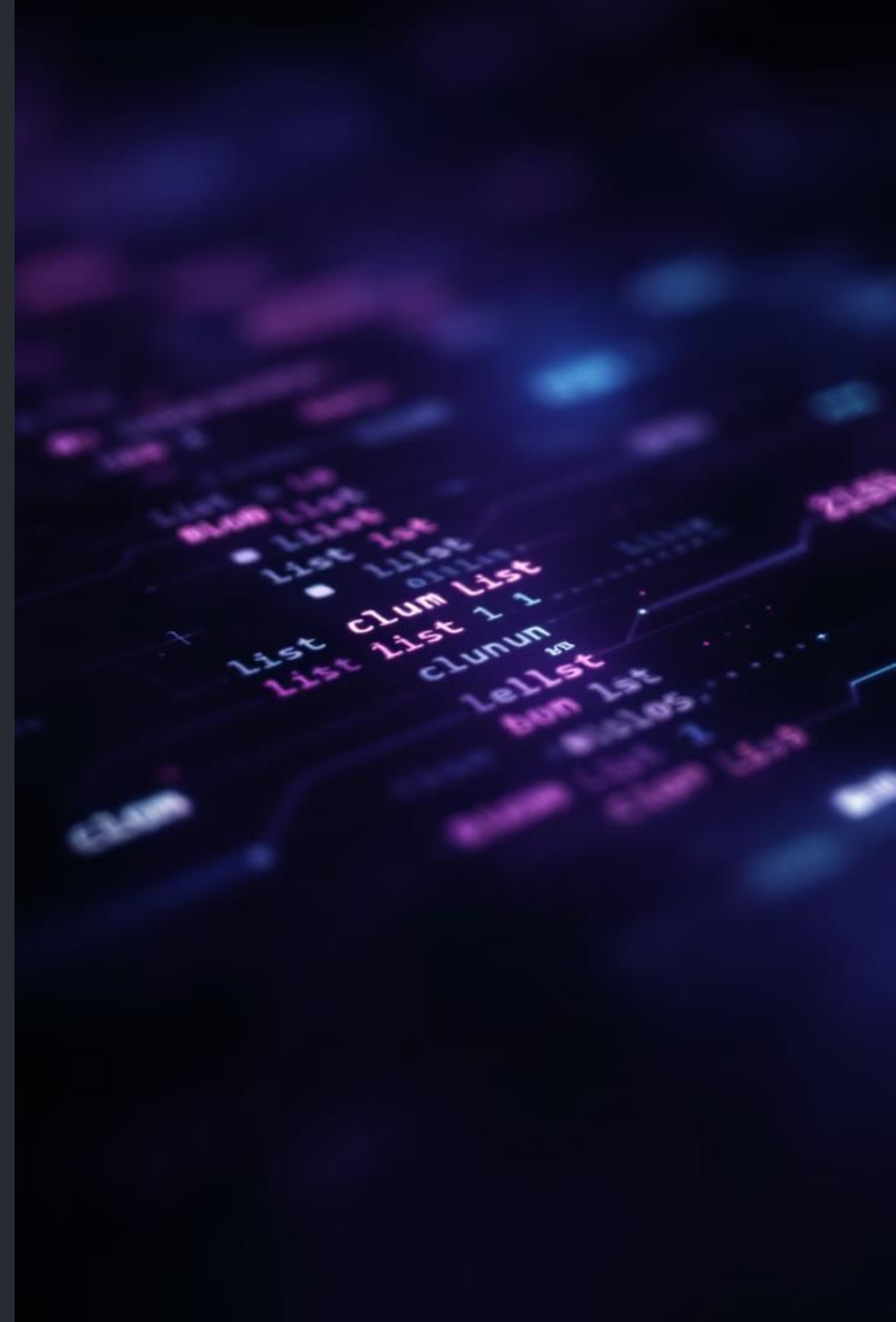
Quando o número de elementos é desconhecido ou varia muito.

2 Flexibilidade Necessária

Quando é necessário
adicionar ou remover
elementos
frequentemente.

3 Gerenciamento Eficiente

Quando o gerenciamento eficiente da memória é importante.



Primeiro Exercício de Implementação: Lista Estática de Inteiros

Implemente uma lista estática de inteiros em C. A lista deve ter um tamanho máximo predefinido. Implemente funções para adicionar, remover e buscar elementos.

Adicionar

Insere um inteiro na lista.

Remover

Remove um inteiro da lista.

Buscar

Encontra um inteiro na lista.



Detalhes e Requisitos do Exercício 1

- Definir um tamanho máximo para a lista (e.g., 10).
- Implementar as funções de adicionar, remover e buscar.
- Tratar erros de lista cheia ou vazia.

1

Tamanho

Defina o tamanho máximo.

2

Funções

Implemente as funções.

3

Tratamento

Trate erros.



Segundo Exercício de Implementação: Lista Dinâmica de Strings

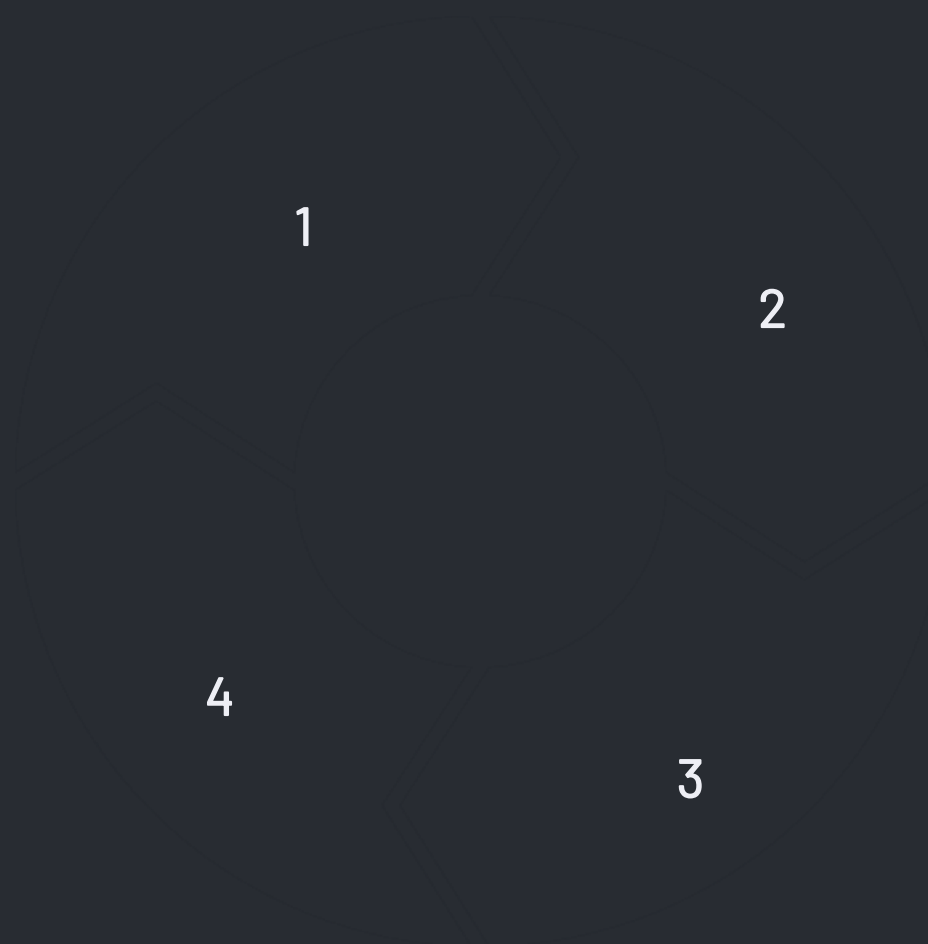
Implemente uma lista dinâmica de strings em C. Use alocação dinâmica de memória para redimensionar a lista conforme necessário. Implemente funções para adicionar, remover e buscar strings.

Alocar
Alocar memória.

Adicionar
Adicionar strings.

Buscar
Buscar strings.

Remover
Remover strings.



Detalhes e Requisitos do Exercício 2

- Usar **malloc**, **calloc** e **free** para gerenciar a memória.
- Implementar funções para adicionar, remover e buscar strings.
- Tratar erros de alocação de memória.

1 Alocação

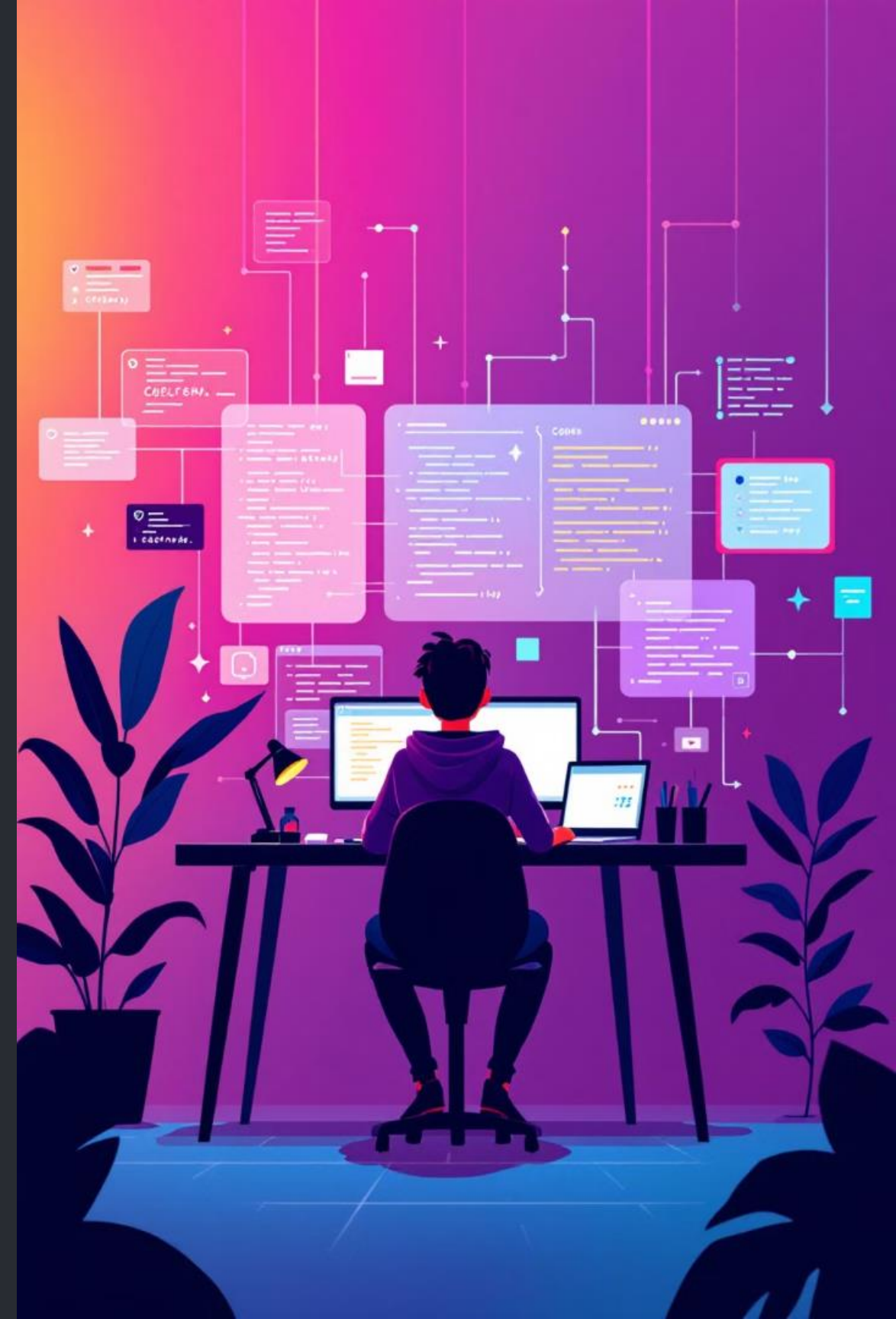
Use malloc/calloc.

2 Funções

Implemente funções CRUD.

3 Erros

Trate erros.





Dicas para Implementação Eficiente

1 Evitar Vazamentos

Sempre liberar a memória alocada com **free**.

2 Alocação Otimizada

Alocar memória em blocos maiores para reduzir overhead.

3 Código Limpo

Usar nomes de variáveis e funções descritivos.

Boas Práticas de Programação em C para Listas

Modularização

Dividir o código em funções menores e reutilizáveis.

Documentação

Comentar o código para explicar o funcionamento.

Testes

Escrever testes unitários para garantir a correção.

WE learning on youmple and
next firee buings nereess,
follow to your browning.

Próximos Passos

- Filas – (FIFO)
- Pilhas – (LIFO)



Perguntas e Respostas

Este é o momento para tirar dúvidas sobre os conceitos e exercícios apresentados. Aproveite para aprofundar seu conhecimento e compartilhar suas experiências.