

Guia de Estudos

Listas Dinâmicas

Lucas Vinícius de Lima Assis

23 de novembro de 2023

Este é um pequeno guia sobre as bases a se estudar para começar a implementar estruturas de dados em C. Para se implementar a mais básica, a Lista Encadeada, é necessário saber fazer uma Alocação Dinâmica, criar Structs e usar Ponteiros.

Este é apenas um guia de estudos, e todo o conhecimento teórico já está disponível em bons livros. Recomendo que usem o [C Completo Total](#). Vão encontrar até mais coisas do que buscavam inicialmente.

Conseguindo fazer esses exercícios básicos creio que possuam tudo o que é preciso para implementar as estruturas de dados

Ponteiros

Este é um código em C que troca os valores de duas variáveis utilizando uma função de troca. No entanto, a função foi apagada. Utilize o conhecimento sobre ponteiros e seus operadores & e * para reescrever uma função de troca funcional.

```
#include <stdio.h>

void troca(int* a, int* b) {
    // Insira seu código aqui
}

int main() {
    int varA = 3;
    int varB = 6;

    printf("%d %d\n", varA, varB);

    troca(&varA, &varB);

    printf("%d %d\n", varA, varB);

    return 0;
}
```

Note que a função `void troca(int* a, int* b)` recebe ponteiros como parâmetros. Isso é realmente necessário? Seria possível fazer uma função `void troca(int a, int b)` que funcione?

Malloc e Free

Este código tem gambiarra! Embora ele funcione como o desejado, perceba que estamos fazendo uma declaração de um vetor estático de maneira "dinâmica". Refatore o código utilizando `malloc` e `free` para que a alocação dinâmica seja realizada corretamente.

```
#include <stdio.h>

int main() {
    int n;
    printf("Digite o n de alunos:\n");
    scanf("%d", &n);

    int alunos[n]; // Isso é GAMBIARRA!

    for (int i=0; i<n; i++) {
        printf("Nota aluno %d: ", i+1);
        scanf("%d", &alunos[i]);
    }

    for (int i=0; i<n; i++) {
        printf("%d ", alunos[i]);
    }
    return 0;
}
```

Structs

Recentemente, comecei a programar um jogo de Digimon em C. Decidi começar simples e meus Digimons vão ter apenas *vida* e *dano*, todos do tipo *int*. Criei uma **struct** para representar os Digimons, e uma função **NovoDigimon**. Essa função aloca dinamicamente o espaço de um Digimon, atribui a vida e o dano neste espaço de memória e me retorna o ponteiro deste Digimon alocado. Por algum motivo eu não salvei a implementação da struct e da função e já até esqueci como tinha feito. Poderia me ajudar?

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

typedef struct Digimon {
    // Esqueci essa struct
} Digimon;

Digimon* NovoDigimon(int vida, int dano) {
    // Esqueci essa função
}

int main() {
    // Digimon com 10 de vida e 15 de dano
    Digimon* digimon1 = NovoDigimon(10, 15);

    printf("Vida %d dano %d",
           digimon1->vida, digimon1->dano);

    return 0;
}
```