ATIVIDADE 3 | LISTAS

Enos Andrade e Alinne Oliveira

1 O que significa alocação sequencial de memória para um conjunto de elementos?

Solução

Numa alocação sequencial, os nós além de estarem em uma sequência lógica, estão também fisicamente em sequência. A maneira prática mais simples de se fazer isso é através da utilização de um vetor.

2 O que significa alocação estática de memória para um conjunto de elementos?

Solução

Em uma alocação estática, uma parte da parte da memória é separada previamente para um determinado fim. Uma vez declarado, esse espaço não poderá ser ampliando, reduzido ou movido de um local para outro em tempo de execução.

3 Qual a diferença entre alocação sequencial e alocação encadeada?

Solução

Em uma alocação sequencial, como dito na solução da primeira questão, os nós estão fisicamente em sequencial e assim, sabendo onde fica o nó inicial, é possível encontrar os nós seguintes. Na alocação encadeada, cada nó além de ter os dados que serão armazenados, têm também o endereço para o nó seguinte.

4 Considere que a struct abaixo está armazenada na sua lista. Faça uma função para buscar o aluno de menor nota. A função deve retornar se a operação foi possível ou não.

```c
struct aluno{
    int mat; //Matricula do aluno
    char nome[10]; //nome do aluno
    float nota; //valor da nota
    int qtdeAlunos; //quantidade de alunos
}
```

Solução

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

typedef struct {
    int mat;//Matricula do aluno
    char *nome;//nome do aluno
    float nota;//valor da nota
} Aluno;

//Lista estática
typedef struct {
    int qtdeAlunos;//quantidade de alunos
    Aluno alunos[30];
} Lista;

int menorNota(Lista *l){
    int posicao = 0;
```

```
    if (l->qtdeAlunos > 0){
        for(int i = 1; i < l->qtdeAlunos; i++){
            if (l->alunos[i].nota < l->alunos[posicao].nota) {
                posicao = i;
            }
        }
        return posicao;
    }
    return -1;
}

int main(){
    Lista *l = (Lista*) malloc(sizeof(Lista));
    l->qtdeAlunos = 4;
    char nomes[4][10]={"Enos", "Alinne", "Irineu", "Danilo"};
    for(int i = 3; i >= 0; i--){
        l->alunos[i].nome = nomes[i];
        l->alunos[i].mat = 20191 + i;
        l->alunos[i].nota = (i+1)*2;
    }

    int posicao = menorNota(l);
    if (posicao != -1) {
        printf("\nAluno com a menor nota \n");
        printf("Nome: %s\n", l->alunos[posicao].nome);
        printf("Matricula: %d\n", l->alunos[posicao].mat);
        printf("Nota: %.1f\n", l->alunos[posicao].nota);

    }else printf("A lista esta vazia");
}
```

5 Considere uma lista contendo números inteiros positivos. Faça uma função que retorne quantos números pares existem na lista.

Solução

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

//Nó da lista
typedef struct {
    int num;
} No;

//Lista estática
typedef struct {
    int qtd;
    No nos[30];
} Lista;

int qtdPares(Lista *l){
    int qtd = 0;

    for(int i = 0; i < l->qtd; i++){
        if (l->nos[i].num % 2 == 0) qtd++;
    }
```

```
    return qtd;
}

int main(){
    Lista *l = (Lista*) malloc(sizeof(Lista));
    l->qtd = 100;
    for(int i = 1; i <= 100; i++){
        l->nos[i].num = i;
    }

    int qtd = qtdPares(l);
    printf("\nQuantidade de numeros pares na lista: %d",qtd);
}
```

6 Implemente uma lista que mostre os 10 melhores alunos da disciplina Estrutura de Dados, turno noite, a estrutura deve conter os dados cadastrais do aluno, as notas e sua respectiva média geral.

Solução

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

typedef struct{
    int matricula;
    char *nome;
    float u1;
    float u2;
    float media;
} Aluno;

typedef struct{
    Aluno alunos[15];
} Lista;


void insertionSort(Lista *l, int n){
    float media;
    Aluno aux;
    for(int i = 1; i < n; i++){
        media = l->alunos[i].media;
        aux = l->alunos[i];
        int j = i - 1;
        while(j >= 0 && l->alunos[j].media < media){
            l->alunos[j + 1] = l->alunos[j];
            j--;
        }
        l->alunos[j + 1] = aux;
    }
}

int main(){
    Lista *l = (Lista*) malloc(sizeof(Lista));
    char nomes[15][20] = {"Dennys Angelim", "Diego Barboza", "Ozivan Brito",
     ↪  "Dalison Carvalho", "Lucas Fernandes", "Severino Gomes", "Samaronia
     ↪  Lacerda", "Edberg Martins", "Felipe Maykon", "Alinne Oliveira", "Tamara
     ↪  Ramalho", "Washington Santos", "Joao Victor", "Joao Pedro", "Enos
     ↪  Andrade"};
```

```
    float u1[15] = {7.1, 7.5, 7.8, 7.4, 8.8, 8.65, 9.5, 8.1, 7.3, 10, 9, 8.7, 7.5,
     ↪  8.5, 10};
    float u2[15] = {8, 8.5, 9, 7.5, 7.1, 8.8, 8.5, 7, 7.7, 9.2, 8.3, 8.9, 9.5, 7,
     ↪  10};
    for(int i = 0; i < 15; i++){
        l->alunos[i].matricula = 2019101 + i;
        l->alunos[i].nome = nomes[i];
        l->alunos[i].u1 = u1[i];
        l->alunos[i].u2 = u2[i];
        l->alunos[i].media = (u1[i] + u2[i]) / 2;
    }
    insertionSort(l, 15);

    for(int i = 0; i < 10; i++){
        printf("\n=========== %do Lugar ===========\nNome: %s\nMatricula:
         ↪  %d\nMedia: %.1f\n", (i+1), l->alunos[i].nome, l->alunos[i].matricula,
         ↪  l->alunos[i].media);
    }
}
```

7 Implemente uma lista que faça uma busca de qualquer um dos alunos da disciplina Estrutura de Dados, turno noite, é interessante mostrar ao usurário uma mensagem, (“Nenhum aluno encontrado”), (“O aluno que você busca é:”)

Solução

Escolhemos não fazer essa.

8 Escreva uma função que conte o número de nós de uma lista encadeada.

Solução

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

struct Nos{
   int num;
   struct Nos *proximo;
};

typedef struct Nos No;

No *lista = NULL;

void inserirPrimeiro(int valor) {
    No *no = (No*) malloc (sizeof(No));
    no->num = valor;
    no->proximo = lista;
    lista = no;
}
void inserirUltimo(int valor) {
    if (lista == NULL) {
        inserirPrimeiro(valor);
        return;
    }
    No *no = lista, *novo;
    while (no->proximo != NULL){
```

```
26          no = no->proximo;
27      }
28      novo = (No*) malloc(sizeof(No));
29      novo->num = valor;
30      novo->proximo = NULL;
31      no->proximo = novo;
32  }
33  int tamanho(){
34      int cont = 0;
35      No *no = lista;
36      while (no != NULL){
37          no = no->proximo;
38          cont++;
39      }
40      return cont;
41  }
42  void main(){
43
44      for(int i = 0; i < 10; i++){
45          inserirUltimo(i+1);
46      }
47      printf("Numero de Nos: %d\n", tamanho());
48  }
```

9 Da lista em que fizemos na sala implemente: Remoção de elementos e comparação de duas listas.

Solução

Escolhemos não fazer essa.

Marque a alternativa correta:

10 _____ são coleções de itens de dados "alinhados em fila" – inserções e exclusões são feitas em qualquer lugar de uma _____.

(a) Listas encadeadas, lista encadeada
(b) Filas, fila
(c) Pilhas, pilha
(d) Árvores binárias, árvore binária

Solução

Alternativa (a)

11 Uma estrutura autorreferente contém um membro _____ que aponta para _____.

(a) inteiro, uma estrutura do mesmo tipo de estrutura
(b) ponteiro, um inteiro
(c) inteiro, um inteiro
(d) ponteiro, uma estrutura do mesmo tipo de estrutura

Solução

Alternativa (d)

12 _____ não é uma vantagem das listas encadeadas em comparação com os arrays.

(a) A alocação dinâmica de memória

(b) A inserção e exclusão eficientes

(c) O acesso direto a qualquer elemento da lista

(d) O uso eficiente da memória

Solução

Alternativa (c)

13 Para uma lista encadeada não vazia, selecione o código que deve aparecer em uma função que acrescenta um nó ao final da lista. newPtr é um ponteiro para o novo nó a ser acrescentado, e lastPtr é um ponteiro para o último nó atual. Cada nó contém um ponteiro, um link para um nó.

(a) lastPtr->nextPtr = newPtr; lastPtr = newPtr;

(b) lastPtr = newPtr; lastPtr->nextPtr = newPtr;

(c) newPtr->nextPtr = lastPtr; lastPtr = newPtr;

(d) lastPtr = newPtr; newPtr->nextPtr = lastPtr;

Solução

Alternativa (a)