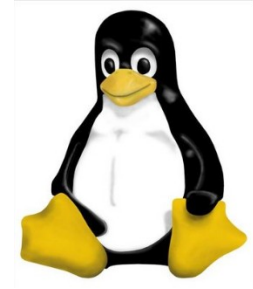


INSTITUTO FEDERAL
BAIANO



LP II

Estrutura de Dados

Listas Lineares Estáticas

Prof. José Honorato Ferreira Nunes

honoratonunes@softwarelivre.org

<http://softwarelivre.org/zenorato/honoratonunes>

Estruturas de Listas

LISTAS LINEARES BÁSICAS

LISTA LINEARES ESTÁTICAS

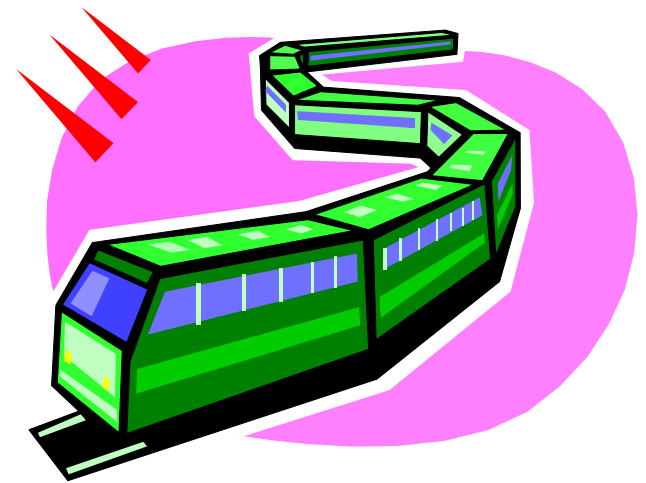
LISTA ENCADEADA CIRCULAR

LISTA DUPLAMENTE ENCADEADA

LISTAS ESPECIALIZADAS

PILHA E FILA

LISTAS ORTOGONAIS E GENERALIZADAS



LISTAS LINEARES ESTÁTICAS

- Nessa aula iremos trabalhar com listas implementadas em vetores.
- Listas lineares estáticas são uma sequência de nós ou elementos dispostos em uma ordem estritamente linear, em que são acessíveis um após o outro, na ordem.
- Pode ser implementado de várias maneiras. Em um simples vetor, na linguagem C, ou em uma estrutura que tem um vetor de tamanho fixo e uma variável inteira para armazenar o tamanho da lista, ou seja a quantidade de elementos do vetor.

LISTAS LINEARES ESTÁTICAS- REPRESENTAÇÃO

Os nós de uma lista são armazenados em endereços contínuos. A relação de ordem entre os nós da lista é representada pelo fato de que se o endereço do nó x_i é conhecido, então o endereço do x_{i+1} pode ser determinado, conforme figura abaixo, onde os endereços dos nós são $k=0$ até $n-1$.

x1	x2	x3	x4	x5	x6	x(n-1)	xn
K=0	1	2	3	4	5	n-2	n-1

LISTAS LINEARES ESTÁTICAS- REPRESENTAÇÃO

A estrutura da lista é um agregado heterogêneo, cuja definição é feita através das seguintes declarações:

```
int m = 10;  
struct lista {  
    int n;  
    char v[m];  
};  
struct lista L;
```

Uma estrutura heterogênea constitui-se de membros cujos tipos são diferentes entre si.

Os membros da estrutura acima são uma variável n como inteiro que serve para armazenar a quantidade de elementos da lista e um vetor de $v[]$ de char que armazena os nós da lista que é do tipo caractere.

LISTAS LINEARES ESTÁTICAS- REPRESENTAÇÃO

Para atribuirmos valores a um membro da lista, devemos utilizar a notação seguinte.

$L.v[0]$ = 'a' – primeiro elemento da lista;

$L.v[1]$ = 'b' – segundo elemento da lista;

$L.v[9]$ = 'z' – décimo elemento da lista.

$L.n = 0$ – a lista está vazia;

$L.n = 10$ – a lista está cheia.

Operações Primitivas

- a) Acessar o k -ésimo nó da lista para obter e ou alterar o dado nele contido;
- b) Inserir um novo nó após o k -ésimo nó da lista;
- c) Inserir um nó no final da lista;
- d) Remover o k -ésimo nó da lista;
- e) Concatenar duas listas;
- f) Determinar o número de nós de uma lista;
- g) Localizar o nó que contém um dado valor;
- h) Gerar uma subcadeia de uma lista (sublista);
- i) Intercalar duas listas;
- j) Inverter uma lista;
- k) Ordenar uma lista.

LISTAS LINEARES ESTÁTICAS-IMPLEMENTAÇÃO

O código abaixo define uma estrutura de lista – L, cujo tamanho máximo é de 10 elementos:

```
#include <stdio.h>
#define m 10
struct Lista {
    int n;
    char elem[m];
};
struct Lista L;
```

a) Iniciar uma lista: antes de inserir algum elemento na lista, devemos primeiro, iniciar a lista.

```
void inicialista(struct Lista *L){
    L->n = 0;
}
```


LISTAS LINEARES ESTÁTICAS-IMPLEMENTAÇÃO

b) Inserir um nó no final da lista.

```
void inserirFinal(struct Lista *L, char dado){
    int i;
    if (L->n > m-1)
        printf("erro - elemento fora dos limites");
    else{
        L->elem[L->n] = dado;
        L->n++;
    }
}
```

LISTAS LINEARES ESTÁTICAS-IMPLEMENTAÇÃO

c) Acessar o k_ésimo nó de uma lista.

```
void acessar(struct Lista *L, int k){
    char dado;
    if ((k < 0 || k > (L->n-1)) || (L->n > m-1))
        printf("erro - elemento fora dos limites");
    else{
        dado = L->elem[k];
        printf("%c", dado);
    }
}
```

Bibliografia Básica

- ❑ SILVA, Osmar Quirino da. **Estruturas de Dados e Algoritmos Usando C – Fundamentos e Aplicações.** Rio de Janeiro, Ed: Ciência Moderna, 2007.
- ❑ MANZANO, Wilson Y. Yamaturni-São Paulo-SP. **Lógica estruturada para programação de computadores,** Ed. Érica 1997 e 2001.
- ❑ MORAES, Celso Roberto. **Estruturas de Dados e Algoritmos.** Ed. Érica, São Paulo.
- ❑ LOPES, Anita. **Introdução à programação.** Rio de Janeiro: Campus, 2002.

Bibliografia Complementar

- ❑ BENEDUZZI, Humberto M. e METZ, João A. **Lógica e Linguagem de Programação – Introdução ao Desenvolvimento de Software** (1ª edição). Editora do Livro Técnico, 2010
- ❑ FORBELLONE, A. L. V. e Eberspacher, H. F. **Lógica de Programação - a Construção de Algoritmos e Estruturas de Dados** (3ª edição). Pearson, 2005
- ❑ CORMEN, Thomas H. et. al. **Algoritmos: Teoria e Prática**. Editora Campus, 2002.
- ❑ ZIVIANI, Nivio. **Projeto de Algoritmos**. Editora Nova Fronteira, 2004.
- ❑ SEBESTA, Robert W. **Conceitos de Linguagens de Programação**. Bookman, 2001.