Campus de Frutal

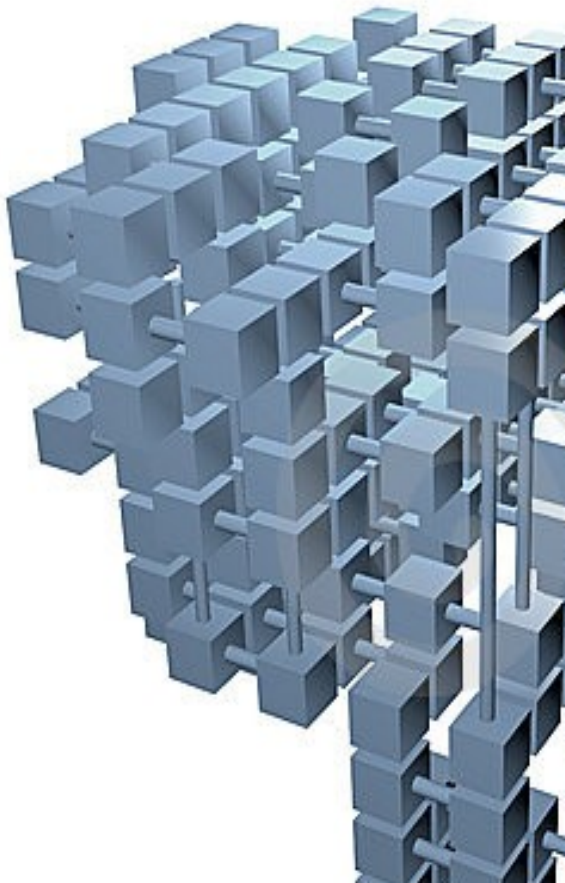
UEMG UNIVERSIDADE DO ESTADO DE MINAS GERAIS

SISTEMAS DE INFORMAÇÃO

Estrutura de Dados 1

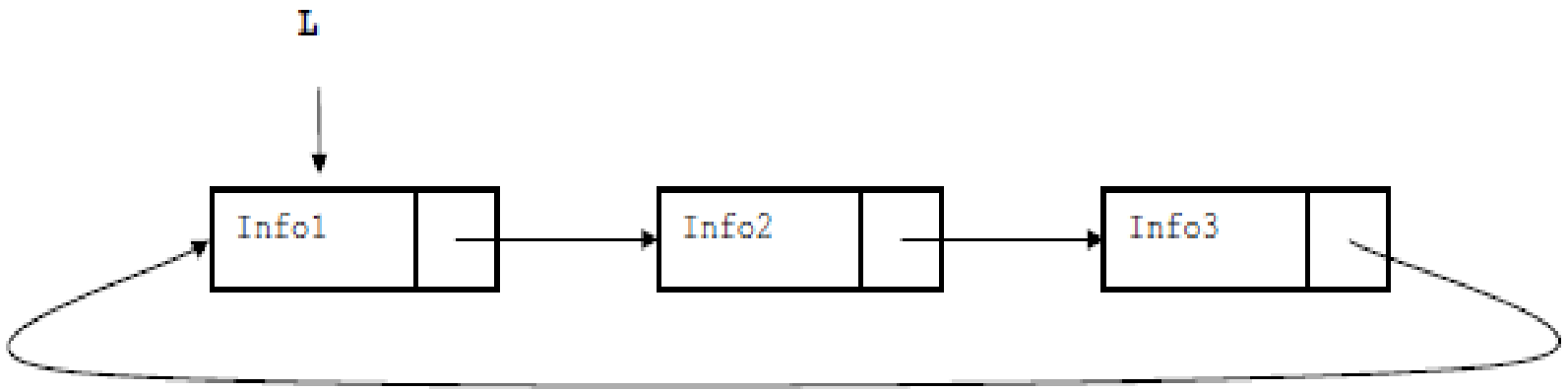
Lista circula e duplamente encadeada

Prof. Ivan José dos Reis Filho
ivanfilhoreis@gmail.com



Lista circular

- O último elemento tem como próximo o primeiro elemento da lista, formando um ciclo
- A lista pode ser representada por um ponteiro para um elemento inicial da lista



Lista circular

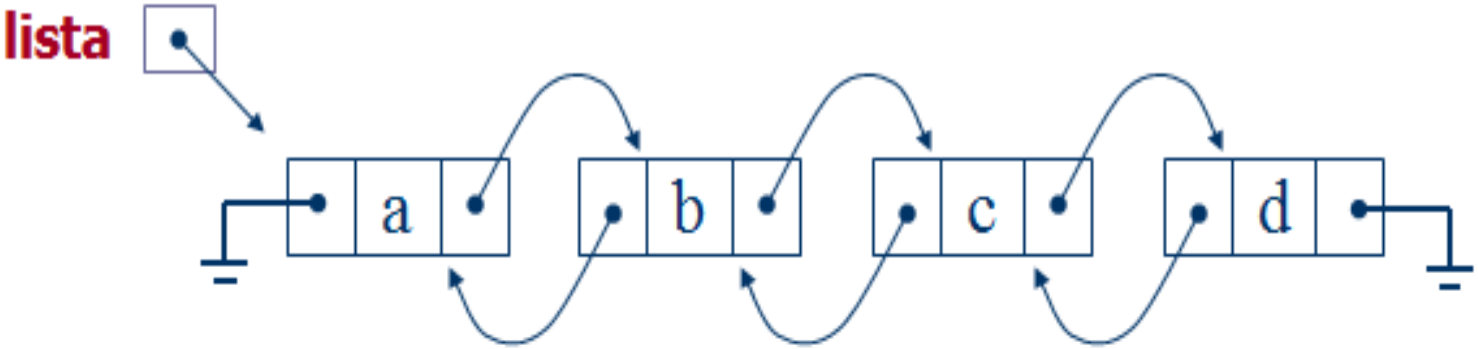
```
/* função imprime: imprime valores dos elementos */
void imprimeCircular (Lista* l)
{
    Lista *p = l; /* faz p apontar para o nó inicial */
/* testa se lista não é vazia e então percorre com do-while*/
    if (p) do
    {
        printf("%d \n", l->info);  /* imprime informação do nó */
        p = p->prox;  /* avança para o próximo nó */
    } while (p != l);
}
```

Lista circular

```
/* inserção no início: retorna a lista atualizada */
Lista *insereLista (Lista *l, int i)
{
    Lista *aux;
    Lista *novo = (Lista*) malloc(sizeof(Lista));
    novo->info = i;
    novo->prox = l;
    if (l==NULL)
        novo->prox = novo;
    else
    {
        aux=l;
    do {
        aux=aux->prox;
        } while (aux->prox!=l);
    aux->prox = novo;
    }
    return novo;
}
```

Lista duplamente encadeada

- Cada nó possui dois ponteiros
  - Um para o elemento anterior e outro para o próximo elemento (ant e prox)



Lista duplamente encadeada

```
/* inserção no início: retorna a lista atualizada */
Lista *insereLista (Lista *l, int i)
{
    Lista *aux;
    Lista *novo = (Lista*) malloc(sizeof(Lista));
    novo.info = i;
    novo.prox = l;
    novo.ant = NULL;
    if(l!=NULL)
        l.ant = novo;
        return novo;
}
```

Lista duplamente encadeada

```
void imprimeInverso (Lista *l)
{
    Lista *p,*a;
    for (p = l; p->prox != NULL; p = p->prox);
    printf("\nImpressao da lista invertida: \n");
    for (a = p; a != NULL; a = a.ant)
    printf("info = %d\n", a->info);
    printf("================== fim da lista inversa");
}
```

Lista duplamente encadeada

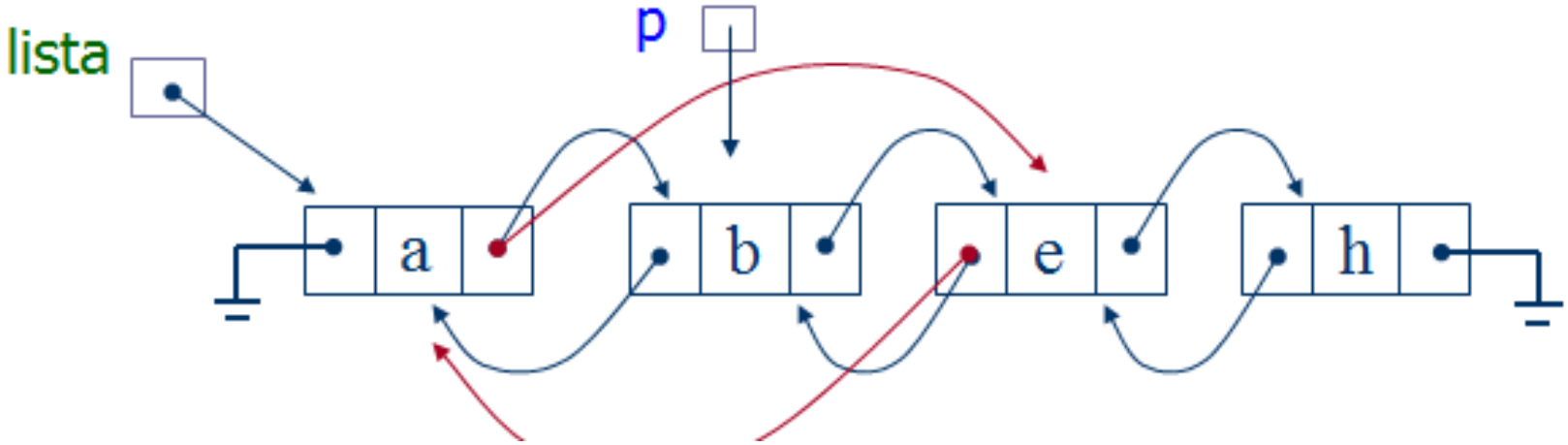
- A remoção é mais trabalhosa, pois é preciso acertar a cadeia nos dois sentidos
- Em compensação, pode-se retirar um elemento conhecendo-se apenas o ponteiro para ele.
- Utiliza-se uma função de busca para localizar um elemento e em seguida o encadeamento é ajustado
- Ao final, o elemento é liberado
- Sendo p o ponteiro para o elemento a ser excluído, se o elemento estiver no meio da lista,

  devemos fazer:

```
p.ant.prox = p.prox;
p.prox.ant = p.ant;
```

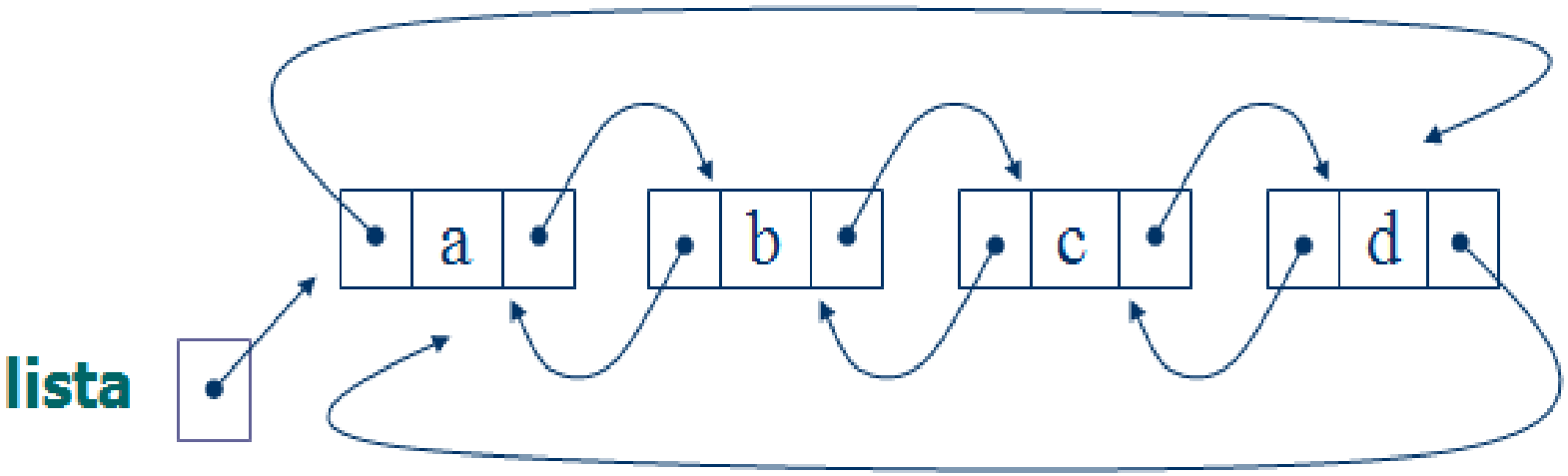
Lista duplamente encadeada

- Caso o elemento esteja no extremo da lista, existem outras condições
  - Se p for o primeiro, não se pode referenciar o p.ant, pois ele é NULL; o mesmo acontece para p.prox quando é o último.
  - Além disso, se for o primeiro, é preciso atualizar o ponteiro da lista



Lista duplamente encadeada circular

- Cada nó possui dois ponteiros: um para o elemento anterior e outro para o próximo elemento (ant e prox)
- O anterior do primeiro é o último e o próximo do último é o primeiro



Trabalho

if (integrantes <= 5 && integrantes >=4)

lista duplamente encadeada circular

lista circular

if (integrantes < 4 && integrantes >=3)

lista estática (vetor)

lista encadeada

if (integrantes <= 2)

lista duplamente encadeada

Trabalho

- Prazo de entrega
  - 24/04 (às 18:00)
- Valor
  - 16,0 Pontos
- Nome dos arquivos
  - Ex:

    ListaEncadeada.c ListaCircular.c

    Listas_NOME_Jose_Francisco_Tiago.rar

Trabalho

- Funções das listas
  - Lista estática
    - Inserir na primeira posição
    - Remover na primeira posição
    - Inserir na última posição
    - Remover na última posição
    - Exibir Elementos da Lista

Trabalho

- Funções das listas
  - Listas Encadeada, Circular, Duplamente encadeada, Duplamente encadeada circular
    - Inserir na primeira posição
    - Remover na primeira posição
    - Exibir elementos da lista

Trabalho

- O trabalho deve conter um MENU
    1. Inserir elemento
    2. Remover elemento
    3. Mostrar elementos da lista
    4. Sair

Grupos

Vamos montar os grupos agora!!!