Universidade de São Paulo - São Carlos,SP
Instituto de Ciências Matemáticas e de Computação

USP - ICMC - SSC
SSC 0300 - 2o. Semestre 2013

Disciplina de
Linguagem de Programação e Aplicações
[Eng. Elétrica / Automação]

Prof. Dr. Fernando Santos Osório / PAE: Rafael Klaser (LRM / ICMC)
LRM - Laboratório de Robótica Móvel do ICMC / CROB-SC
Email: fosorio@icmc. usp. br ou fosorio@ gmail. com
Página Pessoal: http://www.icmc.usp.br/~fosorio/

Material on-line:
Wiki ICMC - http://wiki.icmc.usp.br/index.php
Wiki SSC0300 - http://wiki.icmc.usp.br/index.php/SSC-300-2013(fosorio)

Aula 09

Linguagem de Programação "C"

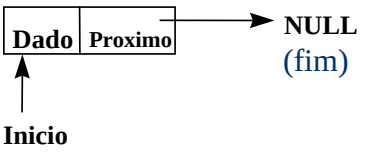
Agenda:

- **Listas Encadeadas Simples: Recordando...**
  - **Fila / Queue (FIFO), Pilha / Stack (LIFO)**
  - **Deque (Double Ended Queue)**
  - **Ponteiros: Inicio / Fim (Head / Tail)** ***Novo conceito!***
- **Listas Encadeadas Duplas:**
  - **Deque**
- **Árvores**
  - **Árvore Binária**
  - **Árvore Binária Ordenada**
  - **Árvore Binária Balanceada**
  - **Árvores Genéricas**
- **Exercícios**

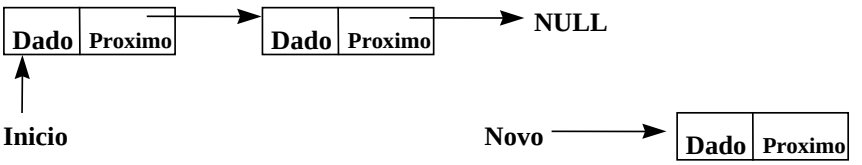
Informações Complementares a Atualizadas:
Consulte REGULARMENTE o material disponível na WIKI
http://wiki.icmc.usp.br/index.php/SSC-300-2013(fosorio)

Listas Encadeada Simples (Ponteiros)

Alocação Dinâmica - Ponteiros: **Lista Encadeada Simples**



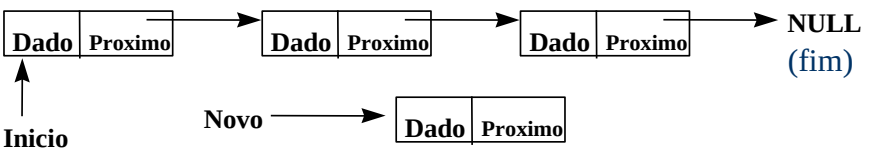
Inserção de Novo Nodo



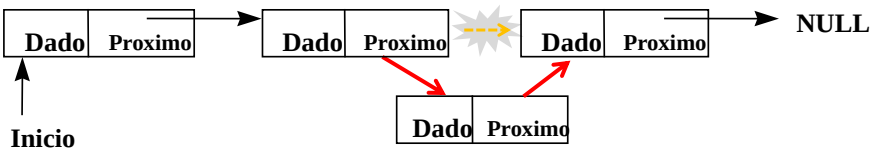
USP – SSC0300 - Semestre 2013-2
Linguagem de Programação e Aplicações
Prof. Fernando Osório / PAE Rafael Klaser

Listas Encadeada Simples (Ponteiros)

Alocação Dinâmica - Ponteiros: **Lista Encadeada Simples**

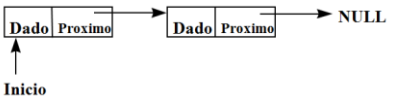


Inserção de Novo Nodo



Listas Encadeada Simples (Ponteiros)

Alocação Dinâmica - Ponteiros: Lista Encadeada Simples



TIPOS DE DADOS:

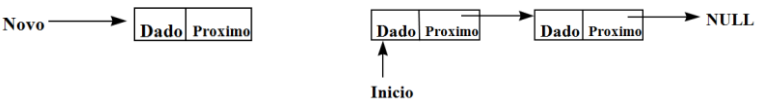
```
typedef  int    Tipo_Dado;

typedef  struct bloco {
            Tipo_Dado      Dado;
            struct bloco  *Proximo;
         }   Nodo;
```

ROTINAS:

```
void inicializa_lista       (Nodo **N);
int insere_inicio_lista     (Nodo **N, Tipo_Dado Dado);
int insere_fim_lista        (Nodo **N, Tipo_Dado Dado);
int insere_ordenando_lista  (Nodo **N, Tipo_Dado Dado);
int remove_inicio_lista     (Nodo **N, Tipo_Dado *Dado);
int remove_fim_lista        (Nodo **N, Tipo_Dado *Dado);
int remove_elemento_lista   (Nodo **N, Tipo_Dado Dado);
int quantidade_lista        (Nodo **N);
void exibe_lista            (Nodo **N);
int pesquisa_lista          (Nodo **N, Tipo_Dado Dado);
int percorre_lista          (Nodo **N, Tipo_Dado *Dado);
void apaga_lista            (Nodo **N);
```


Listas Encadeada Simples (Ponteiros)

Alocação Dinâmica - Ponteiros: Lista Encadeada Simples

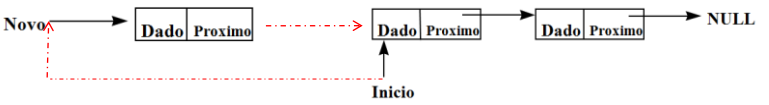


ROTINA INSERIR NO INICIO:

```
int insere_inicio_lista (N, Dado)
Nodo **N;
Tipo_Dado Dado;
{
        Nodo   *novo;

        novo = (Nodo *) calloc ( 1, sizeof (Nodo) );
        if  ( novo == NULL )
              return (ERRO);       /* Não conseguiu alocar na memória */
        novo -> Dado = Dado;
        novo -> Proximo = *N;
        *N = novo;
        return (OK);
}
```

Listas Encadeada Simples (Ponteiros)

Alocação Dinâmica - Ponteiros: Lista Encadeada Simples



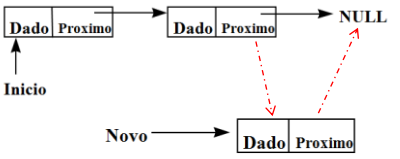
ROTINA INSERIR NO INICIO:

```
int insere_inicio_lista (N, Dado)
Nodo **N;
Tipo_Dado Dado;
{
        Nodo   *novo;

        novo = (Nodo *) calloc ( 1, sizeof (Nodo) );
        if  ( novo == NULL )
              return (ERRO);       /* Não conseguiu alocar na memória */
        novo -> Dado = Dado;
        novo -> Proximo = *N;
        *N = novo;
        return (OK);
}
```


Listas Encadeada Simples (Ponteiros)

Alocação Dinâmica - Ponteiros: Lista Encadeada Simples

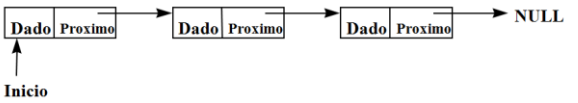
ROTINA INSERIR NO FINAL:




```
int insere_fim_lista (N, Dado)
Nodo **N;
Tipo_Dado Dado;
{
    Nodo  *aux,  *novo;

    novo = (Nodo *) calloc ( 1, sizeof (Nodo) );
    if  ( novo == NULL ) return(ERRO);
    novo -> Dado = Dado;
    novo -> Proximo = NULL;

    if ( *N == NULL )   /* 1o. da lista? */
         *N = novo;
    else  {
             aux = *N;
             while ( aux -> Proximo  !=  NULL )
                      aux = aux -> Proximo;
             aux -> Proximo = novo;
          }
        return (OK);
}
```

Listas Encadeada Simples (Ponteiros)

Alocação Dinâmica - Ponteiros: **Lista Encadeada Simples**

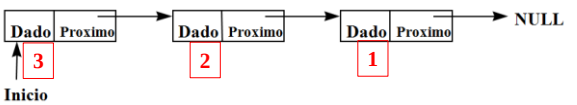


ROTINA EXIBIR LISTA:

```
int exibe_lista (N)
Nodo **N;
{
    Nodo  *aux;

    aux = *N;
    if (aux != NULL)
    {
        while ( aux !=  NULL )
        {
           printf("Dado: %d\n",aux->Dado);
           aux = aux -> Proximo;
        }
    }
    else   printf("Lista vazia!\n");
}
```


Listas Encadeada Simples (Ponteiros)

Alocação Dinâmica - Ponteiros: **Lista Encadeada Simples**




```
#include "listsimp.c"

main()
{
  Nodo *Inicio;

  printf("\n ROTINAS DE MANIPULACAO DE LISTAS SIMPLES ENCADEADAS \n\n");

  inicializa_lista(&Inicio);
  insere_inicio_lista(&Inicio,1);
  insere_inicio_lista(&Inicio,2);
  insere_inicio_lista(&Inicio,3);

  exibe_lista(&Inicio);

  system("pause");
}
```

Listas Encadeada Simples: Início e Fim

Listas Encadeadas Simples:

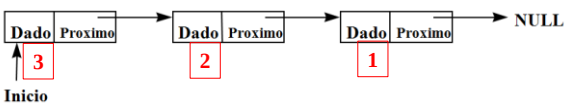
- Podemos trabalhar com ponteiros auxiliares
  INICIO e FIM
  "Head / Cabeça" e "Tail / Cauda"

- Insere no Final:
  Precisava percorrer toda a lista CADA vez que fosse inserir um dado no final da lista!

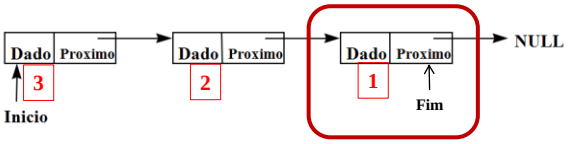
Listas Encadeada Simples (Ponteiros)

Alocação Dinâmica - Ponteiros: **Lista Encadeada Simples**

Insere no INÍCIO



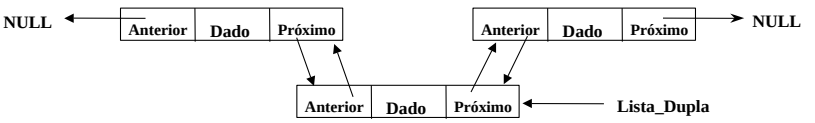
Insere no FINAL



Listas Encadeada DUPLA (Ponteiros)

Alocação Dinâmica - Ponteiros: Lista Encadeada DUPLA

Duplamente Encadeada



Estrutura de dados:

```
typedef    int    Tipo_Dado;

typedef    struct  bloco_ld  {
                                    Tipo_Dado          Dado;
                                    struct bloco_ld    *Proximo, *Anterior;
                           }   Nodo_LD;
```


USP – SSC0300 - Semestre 2013-2
Linguagem de Programação e Aplicações
Prof. Fernando Osório / PAE Rafael Klaser

Listas Encadeada DUPLA (Ponteiros)

Alocação Dinâmica - Ponteiros: Lista Encadeada DUPLA

Duplamente Encadeada

Rotinas:

```
void inicializa_ld          (Nodo_LD **LD);
void posiciona_inicio_ld    (Nodo_LD **LD);
void posiciona_fim_ld       (Nodo_LD **LD);

int insere_inicio_ld        (Nodo_LD **LD, Tipo_Dado Dado);
int insere_fim_ld           (Nodo_LD **LD, Tipo_Dado Dado);
int insere_antes_ld         (Nodo_LD **LD, Tipo_Dado Dado);
int insere_depois_ld        (Nodo_LD **LD, Tipo_Dado Dado);
int insere_ordenando_ld     (Nodo_LD **LD, Tipo_Dado Dado);

int pesquisa_ld             (Nodo_LD **LD, Tipo_Dado Dado);
int remove_nodo_ld          (Nodo_LD **LD, Tipo_Dado *Dado);
int remove_dado_ld          (Nodo_LD **LD, Tipo_Dado *Dado);
int quantidade_ld           (Nodo_LD *LD);
void exibe_ld               (Nodo_LD *LD);
int percorre_lista          (Nodo_LD **LD, Tipo_Dado *Dado);
void apaga_ld               (Nodo_LD **LD);
```

Listas Encadeada DUPLA (Ponteiros)

Alocação Dinâmica - Ponteiros: Lista Encadeada DUPLA

Duplamente Encadeada

Exemplo: posiciona o ponteiro no início/fim da lista duplamente encadeada

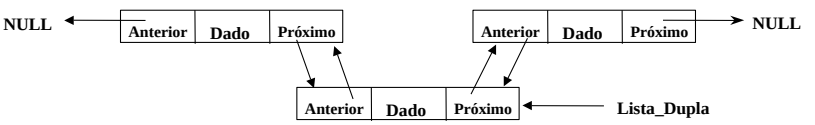
```
void posiciona_inicio_ld (LD)
Nodo_LD   **LD;
{
        if  ( *LD != NULL )   {
            while (  (*LD) -> Anterior != NULL )
                    *LD = (*LD) -> Anterior;
        }
}

void posiciona_fim_ld (LD)
Nodo_LD   **LD;
{
        if  ((*LD != NULL) && (( *LD)->Proximo != NULL ))   {
            while (  (*LD) -> Proximo != NULL )
                    *LD = (*LD) -> Proximo;
        }
}
```


USP – SSC0300 - Semestre 2013-2
Linguagem de Programação e Aplicações
Prof. Fernando Osório / PAE Rafael Klaser

Listas Encadeada DUPLA (Ponteiros)

Alocação Dinâmica - Ponteiros: Lista Encadeada DUPLA

Duplamente Encadeada



Rotinas:

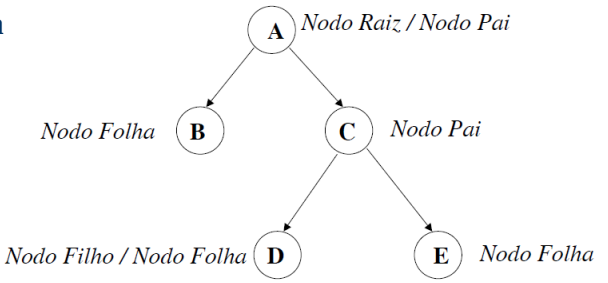
Fila, Pilha, Lista, Deque, Ordenado, Qualquer Ordem

```
int insere_inicio_ld        (Nodo_LD **LD, Tipo_Dado Dado);
int insere_fim_ld           (Nodo_LD **LD, Tipo_Dado Dado);
int insere_antes_ld         (Nodo_LD **LD, Tipo_Dado Dado);
int insere_depois_ld        (Nodo_LD **LD, Tipo_Dado Dado);
int insere_ordenando_ld     (Nodo_LD **LD, Tipo_Dado Dado);
```

Árvore (Ponteiros)

Alocação Dinâmica - Ponteiros: ÁRVORE

Árvore Binária



Estrutura de dados:

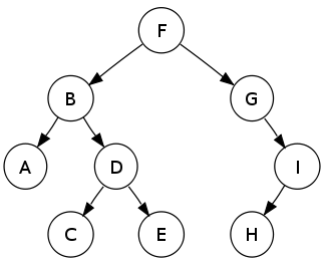
```
typedef int Tipo_Dado;

typedef struct bloco_ab {
                Tipo_Dado Dado;
                struct bloco_ab *FilhoEsq, *FilhoDir;
                struct bloco_ab *Pai; /* opcional */
              } Nodo_AB;
```

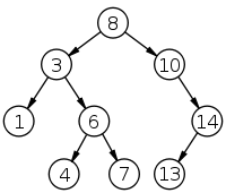

USP – SSC0300 - Semestre 2013-2
Linguagem de Programação e Aplicações
Prof. Fernando Osório / PAE Rafael Klaser

Árvore (Ponteiros)

Alocação Dinâmica - Ponteiros: ÁRVORE

Árvore Binária ORDENADA





ESQ: Menores
DIR: Maiores

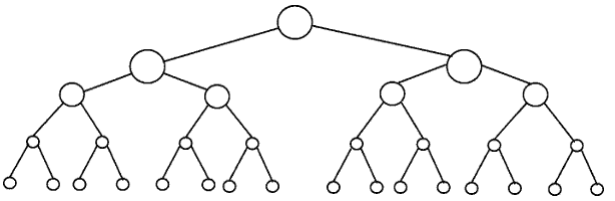
Estrutura de dados:

```
typedef int Tipo_Dado;

typedef struct bloco_ab {
                Tipo_Dado Dado;
                struct bloco_ab *FilhoEsq, *FilhoDir;
                struct bloco_ab *Pai; /* opcional */
              } Nodo_AB;
```

Árvore (Ponteiros)

Alocação Dinâmica - Ponteiros: **ÁRVORE**

Árvore Binária ORDENADA BALANCEADA



Estrutura de dados:

typedef int Tipo_Dado;

typedef struct bloco_ab {
Tipo_Dado Dado;
struct bloco_ab *FilhoEsq, *FilhoDir;
struct bloco_ab *Pai; /* opcional */
} Nodo_AB;

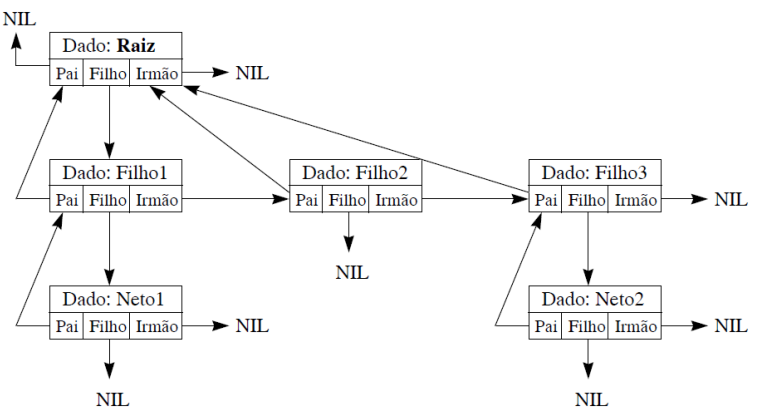
ESQ: Menores
DIR: Maiores

USP – SSC0300 - Semestre 2013-2
Linguagem de Programação e Aplicações
Prof. Fernando Osório / PAE Rafael Klaser

Árvore (Ponteiros)

Alocação Dinâmica - Ponteiros: **ÁRVORE**

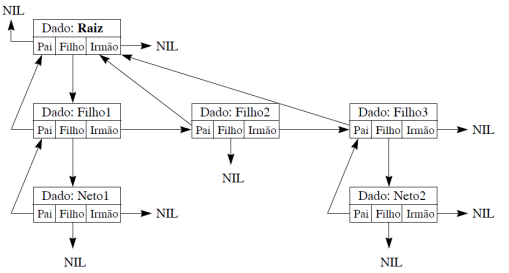
Árvore GENÉRICA



Árvore (Ponteiros)

Alocação Dinâmica - Ponteiros: ÁRVORE

Árvore GENÉRICA



Estrutura de dados:

```
typedef int Tipo_Dado;

typedef struct bloco_ag {
        Tipo_Dado Dado;
        struct bloco_ag *Pai, *Filho, *Irmao;
} Nodo;
```


Universidade de São Paulo - São Carlos,SP
Instituto de Ciências Matemáticas e de Computação

INFORMAÇÕES SOBRE A DISCIPLINA

USP - Universidade de São Paulo - São Carlos, SP
ICMC - Instituto de Ciências Matemáticas e de Computação
SSC - Departamento de Sistemas de Computação

Prof. Fernando Santos OSÓRIO
Web institucional: http://www.icmc.usp.br/
Página pessoal: http://www.icmc.usp.br/~fosorio/
Página do Grupo de Pesquisa: http://www.lrm.icmc.usp.br/
E-mail: fosorio [at] icmc. usp. br ou fosorio [at] gmail. com

Disciplina de Linguagem de Programação e Aplicações SSC300
WIKI - http://wiki.icmc.usp.br/index.php/SSC-300-2013(fosorio)
> Programa, Material de Aulas, Critérios de Avaliação,
> Trabalhos Práticos, Datas das Provas, Notas