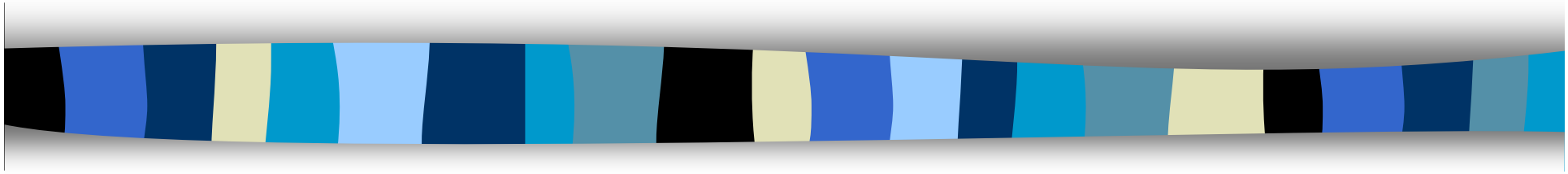
SCC122 – Estruturas de Dados



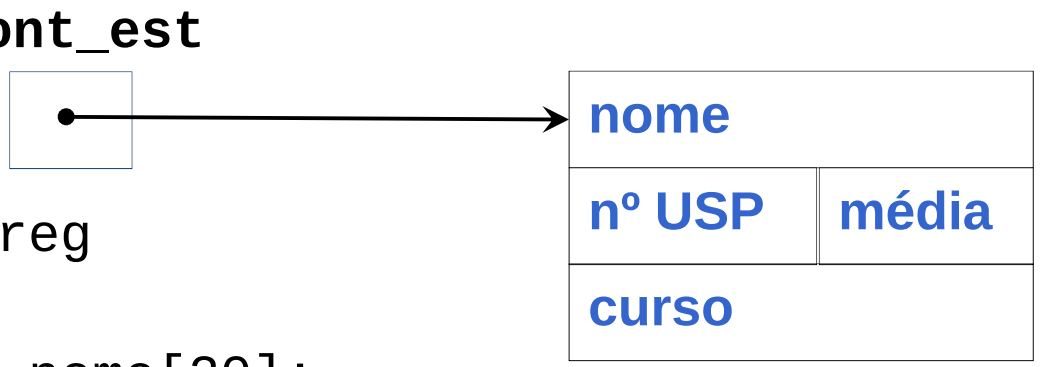
Variável Dinâmica
Lista Encadeada Dinâmica

Variável Dinâmica

- Uma variável dinâmica é uma variável criada (e destruída) explicitamente durante a execução do programa.
- **Objetivo**: Otimizar a utilização da MP.
- Uma variável dinâmica não é declarada, pois ela inexiste antes da execução. A referência a ela é feita através de uma variável **ponteiro** que contém o valor do endereço da **variável dinâmica**.
- A variável **ponteiro** deve ser declarada.
- Ao declarar uma variável **ponteiro**, deve-se declarar o tipo de valor para o qual ela aponta.

Variável Dinâmica: Exemplo

- Suponha que uma variável dinâmica deverá conter os dados de um estudante (*nome, nº USP, curso, média*).

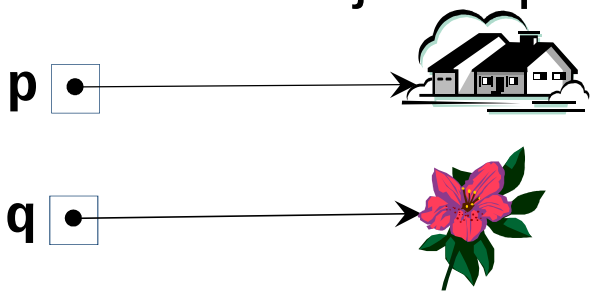



```
    struct treg
    {
        char nome[20];
        int n_usp;
        float media;
        cc curso;
    };
treg *pont_est;
```

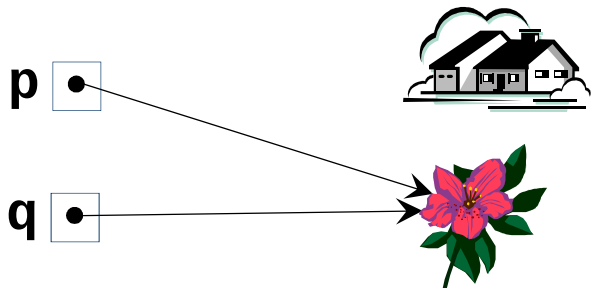
Variável Dinâmica: Exemplo

`p ,q são ponteiros para objetos; (*p , *q)`

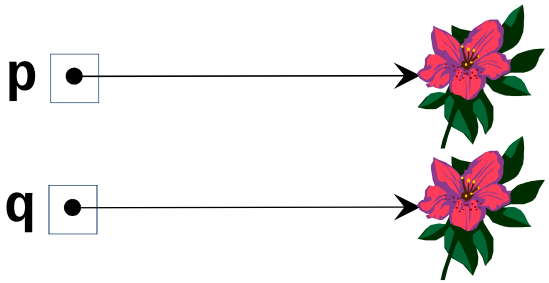
Ponteiro Objeto Apontado



`p = q`

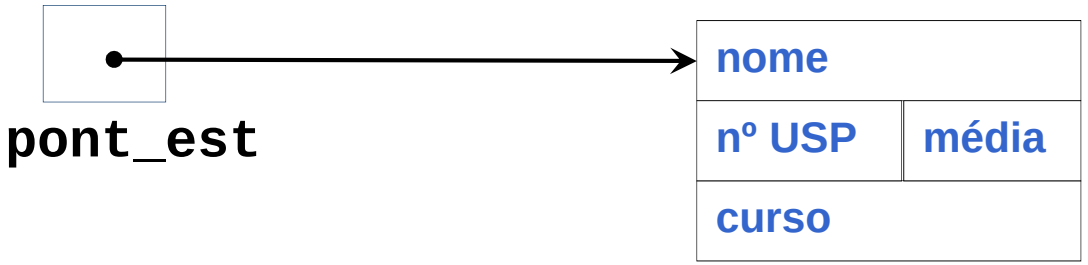


`*p = *q`



Criação da Variável Dinâmica

- **`pont_est =(treg *) malloc(sizeof(treg))`**
- aloca memória para um dado do tipo **`treg`** e estabelece a ligação:

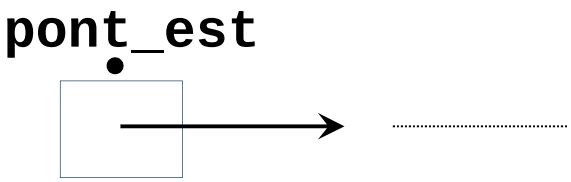


- Para acessar os campos:

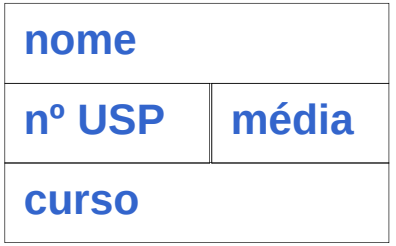
```
pont_est->nome
pont_est->n_usp
pont_est->media
pont_est->curso
```

Destruindo uma Variável Dinâmica

- Para liberar o espaço de memória que a variável dinâmica ocupa:
- **`free(pont_est);`**



`inacessível`

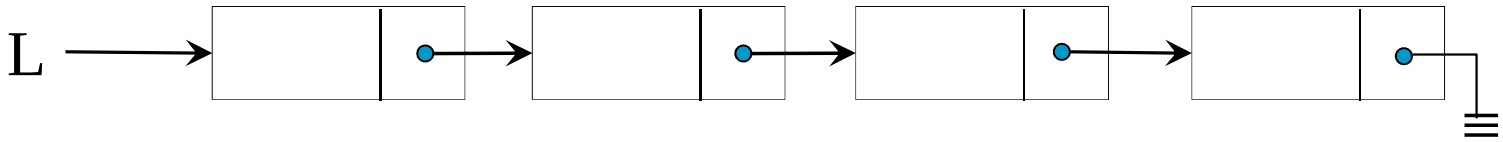


- A constante pré-definida **`NULL`** indica ligação nula.

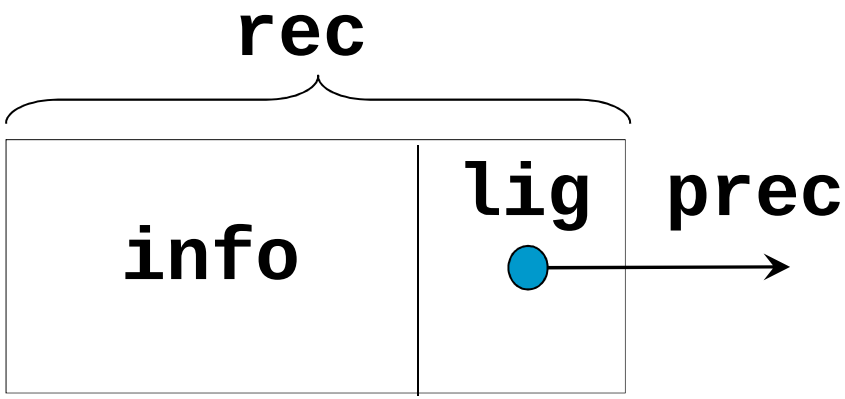
Lista Encadeada Dinâmica

- Os registros disponíveis para **inserção** correspondem a toda memória disponível para o programa durante a execução.
- O campo de ligação dos registros não contem mais índices de array, mas endereços reais da memória principal.
- Quem controla esses endereços é o SO através de comandos da linguagem de programação.
- **C** manipula registro da memória principal dinamicamente (em tempo de execução) através do tipo de dados **ponteiro->**

Lista Encadeada Dinâmica

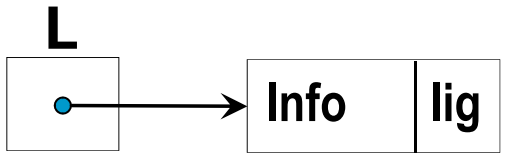



```
struct no_lista
{
        tipo_elem info;
        no_lista *lig;
};
No_lista *L;
```



Principal Utilização de Ponteiros

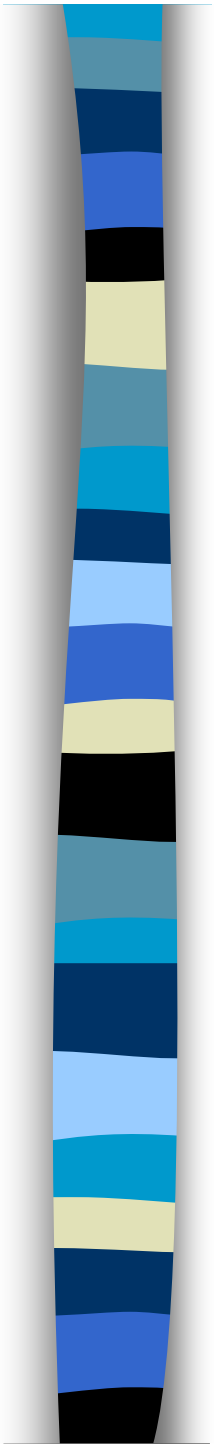
- Listas Encadeadas



- Para designar
  - Ponteiro: `L`
  - Objeto apontado: `*L`
  - Campo do Objeto: `L->info; L->lig;`
  - Ligação nula (lista vazia): `null`

Exemplo: `If L==null then` ... {se lista vazia} Ou

`If L->lig == null then` ... {último}



- Como obter um registro dinamicamente do tipo **rec**?

```
no_lista *p;
P = (no_lista *)
malloc(sizeof(no_lista);
```

- Como devolver dinamicamente **p**?

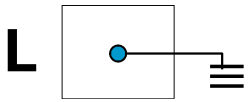
```
free(p);
```

O espaço de **p** volta a fazer parte da MP disponível.

Operações sobre Listas Dinâmicas

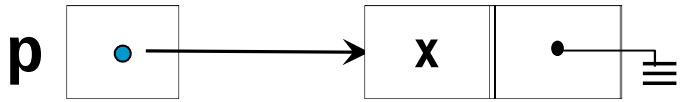
- Criação de uma lista vazia

```
L = null;
```



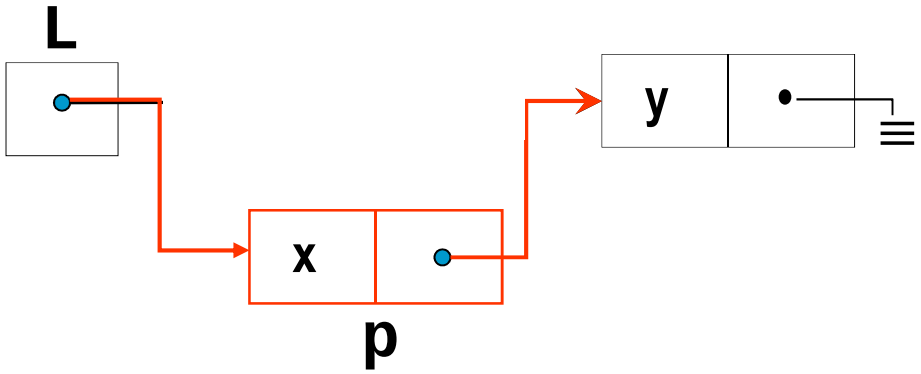
- Inserção do 1º elemento.

```
P =(no_lista *)malloc(sizeof(no_lista));
P->info = x;
P->lig = null;
```



- Inserção no início de qualquer lista.

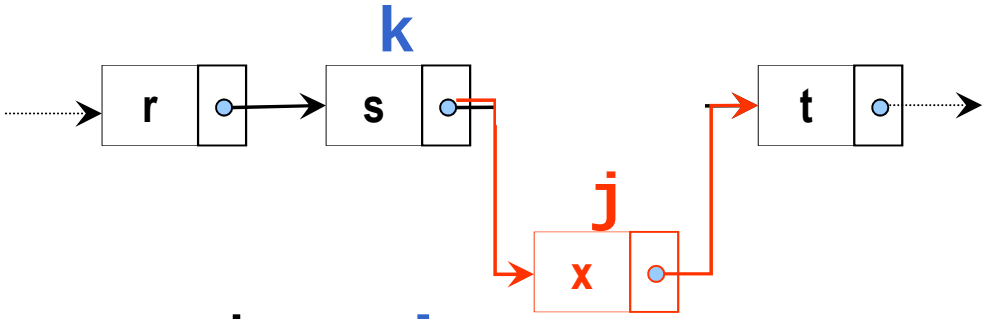
```
P =(no_lista *) malloc(sizeof(no_lista));
P->info = x;
P->lig = L;
L = p;
```



Operações sobre Listas Dinâmicas

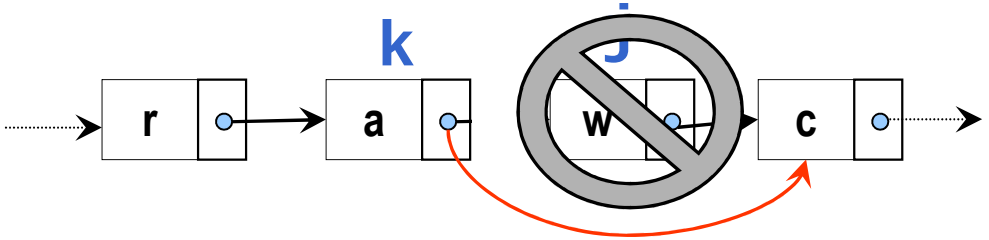
- Inserção após o registro **k**

```
J = (no_lista *)malloc(sizeof(no_lista));
J->info = x;
J->lig = k->lig;
K->lig = j;
```



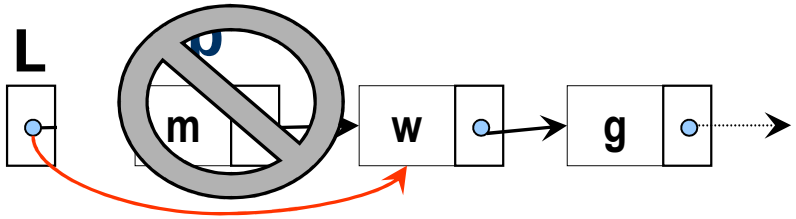
- Eliminação após o registro **k**

```
j = k->lig;
K->lig = j->lig;
free(j);
```



- Eliminação do 1º elemento da lista

```
p = L;
L = L->lig;
free(p);
```



Operações sobre Listas Dinâmicas

- Acesso a um registro da lista (percurso na lista)

```
p = L;
while (p != null)
{
  efetua_calc(p);
  p = p->lig;
};
```

- Versão recursiva.

```
void VISITA_NO(no_lista *p);
{
   if (p != null)
   {
     efetua_calc(p);
     VISITA_NO(p->lig);
   };
};
```

Exercícios

1. Refazer os procedimentos de inserção, eliminação e busca para listas encadeadas dinamicamente e ordenadas pelas chaves.

2. Criar uma classe para listas encadeadas dinâmicas.

- Estes slides foram elaborados pela profa. Roseli Romero com o auxílio dos alunos PAE: Gedson Faria, Rodrigo Silva Duran e Janderson Rodrigo de Oliveira.