ALGORITMOS E ESTRUTURAS DE DADOS I

Prof. Caio César de Freitas Dantas

Alocação Dinâmica de Memória

Utilizado quando se deseja alocar espaço para variável em tempo de execução.

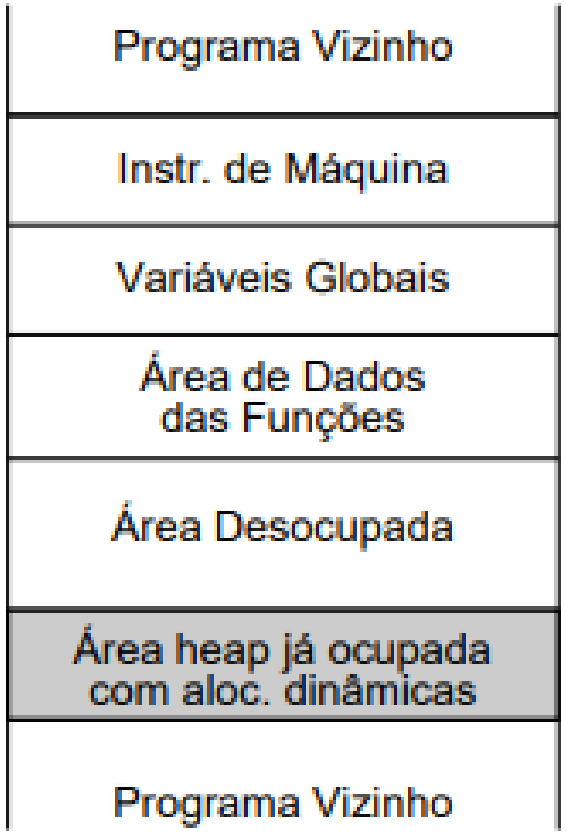
Utilizado principalmente para variáveis indexadas (vetores e matrizes) e estruturas.
A variável deve ser do tipo ponteiro.

A reserva de memoria pode ser feita pelo uso da função malloc(), pertencente a biblioteca stdlib.h.

Essa função recebe como parâmetro o numero de bytes a ser reservado.

Alocação Dinâmica de Memória

A reserva é feita nessa região, ocupando o numero de bytes passado pela função, em endereços contíguos.



Alocação Dinâmica de Memória

```
int main() {
int i, m, A[300], B[300], C[300];

printf("Entre tamanho dos vetores: ");
scanf("%d", &m);

printf("\nVetor A: ");
for (i = 0; i < m; i++)
scanf("%d", &A[i]);

printf("\nVetor B: ");
for (i = 0; i < m; i++)
scanf("%d", &B[i]);
```

Alocação Dinâmica de Memória

- Observe que foram utilizadas 900 posições de memoria antes mesmo que o valor de m seja conhecido.
- Se m = 2, por exemplo, estamos ”desperdiçando” 894 posições.
- A alocação dinâmica permite criar o vetor somente após conhecermos seu real tamanho.

Alocação Dinâmica de Memória

```
int main() {
int i, m, *A, *B, *C;

printf("Entre tamanho dos vetores: ");
scanf("%d", &m);

A = (int *) malloc(m * sizeof(int));
B = (int *) malloc(m * sizeof(int));
C = (int *) malloc(m * sizeof(int));

printf("\nVetor A: ");
for (i = 0; i < m; i++)
scanf("%d", &A[i]);
```

Alocação Dinâmica de Memória

Na atribuição:

```
A = (int *) malloc (m * sizeof(int));
```

- A função malloc() reserva um espaço contíguo de m*sizeof(int) bytes e retorna o endereço inicial desse espaço.
- Assim, a partir desse momento, a variável A passa a atuar como uma variável indexada (vetor) comum.
- Por fim, a função free() libera a memoria reservada é apontada pelo ponteiro passado como argumento.

Alocação Dinâmica de Memória

Estruturas também podem ser alocadas dinamicamente.

Por exemplo, considere a seguinte estrutura:

```
typedef struct st st;
struct st {
int a;
float b;
};

st *p;
```

O comando p = (st *) malloc (sizeof(st)); aloca espaço para uma estrutura st, cujo endereço é atribuído à variável p.

Alocação Dinâmica de Memória

As seguintes atribuições podem ser feitas aos campos dessa estrutura:

p->a = 2; p->b = 3.4;

```
typedef struct st st;
struct st {
int a;
float b;
};

st *p;
```

Alocação Dinâmica de Memória

1- Faça um programa que leia o tamanho de um vetor de inteiros e reserve dinamicamente memoria para esse vetor. Em seguida, leia os elementos desse vetor, imprima o vetor lido e mostre o resultado da soma dos números ímpares presentes no vetor.

Alocação Dinâmica de Memória

2- Criar uma estrutura que represente uma pessoa, contendo nome, data de nascimento e CPF. Aloque dinamicamente uma variável desse novo tipo de dado. Depois crie uma função que receba este ponteiro e preencha os dados da estrutura. A seguir crie também uma função que receba este ponteiro e imprima os dados da estrutura. Finalmente, faça a chamada a esta função na função principal.

Alocação Dinâmica de Memória

3- Faça um programa que leia o tamanho de dois vetores (A e B). Em seguida, leia os elementos desses vetores, imprima um terceiro vetor C contendo os maiores elementos de cada índice dos vetores A e B. Deve-se Alocar dinamicamente espaço para todos os vetores.

FIM!