

Programação II

Alocação Dinâmica

Jocélio Passos
joceliopassos@bol.com.br

02/02/2007 Ir p/ primeira página

Alocação Dinâmica - Conceito

■ Estrutura da Memória Principal RAM



02/02/2007

Baixa
2

02/02/2007 Ir p/ primeira página

Alocação Dinâmica - Conceito

- C Comp Total, Cap 16 (420 - 441)
- Armazenamento em memória principal
 - ◆ Variáveis, vetores, matrizes e struct
 - Alocação Estática
 - Armazenamento fixo em tempo de execução
 - Alocado do Espaço da **pilha**
 - Devemos saber e determinar previamente quanto de espaço cada variável ocupará

02/02/2007

3

02/02/2007 Ir p/ primeira página

Alocação Dinâmica - Conceito

- Mas quando precisamos ajustar a necessidade de armazenamento em tempo de execução ?
- Usamos a Alocação Dinâmica :
- Podemos criar estruturas de dados dinâmicas, cujo tamanho será determinado em tempo de execução (alocar).
- Após seu uso a estrutura deve ser desalocada.
- O espaço necessário ao programa é alocado da memória livre ou **heap**

02/02/2007

4

02/02/2007 Ir p/ primeira página

Alocação Dinâmica - Funções

- Biblioteca : **STDLIB.H**
- `void *malloc (size_t size) :`
 - ◆ aloca um espaço da memória livre
 - ◆ Utiliza ponteiro para manipular o espaço alocado
 - ◆ A quantidade de memória (size) deve ser especificada
 - ◆ Caso não haja memória suficiente um ponteiro nulo é devolvido
 - ◆ Por isso este ponteiro deve ser sempre testado na alocação
- `void free (void *ptr)`
 - ◆ libera a memória alocada

02/02/2007

5

02/02/2007 Ir p/ primeira página

Alocação Dinâmica - sizeof

- Para alocar memória precisamos saber quando espaço cada tipo ocupa
- O operador em tempo de compilação **sizeof** nos dá estes valores
- C Comp Total, Cap 2 (53-54, 194-196)
- Faça um programa que mostre o tamanho em byte que cada tipo de dados abaixo ocupa na memória :
 - ◆ char
 - ◆ short
 - ◆ int
 - ◆ float
 - ◆ double
- Exemplo : `printf ("%d byte", sizeof(char)) // 1 byte`

02/02/2007

02/02/2007 Ir p/ primeira página

Alocação Dinâmica - sizeof

- #include <stdio.h>
- #include <stdlib.h>
- #include <conio.h>
- main()
- { char texto[10];
- short idades[10];
- int indices[10];
- float porcentagens[10];
- double fatores[10];

02/02/2007

7

Ir p/ primeira página



Alocação Dinâmica - sizeof

- printf ("\n\nDado char: %d byte", sizeof(char));
- printf ("\nEspaco para 'char texto [10]': %d bytes",sizeof(texto));
- printf ("\n\nDado short: %d bytes",sizeof(short));
- printf ("\nEspaco para 'short idades [10]': %d bytes",sizeof(idades));
- printf ("\n\nDado int: %d bytes",sizeof(int));
- printf ("\nEspaco para 'int indices [10]': %d bytes",sizeof(indices));

02/02/2007

8

Ir p/ primeira página



Alocação Dinâmica - sizeof

- printf ("\nEspaco para 'int indices [10]': %d bytes",sizeof(indices));
- printf ("\n\nDado float: %d bytes",sizeof(float));
- printf ("\nEspaco para 'float porcentagens [10]': %d bytes",sizeof(porcentagens));
- printf ("\n\nDado double: %d bytes", sizeof(double));
- printf ("\nEspaco para 'double fatores [10]': %d bytes",sizeof(fatores));
- getch();
- }

02/02/2007

9

Ir p/ primeira página



Alocação Dinâmica - sizeof

- Então :

Tipo	Tamanho (em bytes)
char	1
short	2
int	2
float	4
double	8

02/02/2007

10

Ir p/ primeira página



Alocação Dinâmica - Exercício

- 1) Crie um programa que leia n idades (inteiras), armazene –as e depois mostre-as. O valor de n será informado pelo usuário. Crie funções e procedimentos para realizar as tarefas.

02/02/2007

11

Ir p/ primeira página



Alocação Dinâmica - Exercício

- #include <stdio.h>
- #include <stdlib.h>
- #include <conio.h>
- int criarvetor (int n);
- void ler vetor (int n);
- void mostrarvetor (int n);
- int *idades; // ponteiro utilizado na alocação

02/02/2007

12

Ir p/ primeira página



Alocação Dinâmica - Exercício

```
int criarvetor (int n)
{
    int result = 1;
    idades = (int *) malloc ((n)* sizeof(int)); // alocação
    if (idades == NULL)
    {
        printf ("\nErro reservando memoria");
        result = 0;
    }
    return result;
}
```

02/02/2007

13

[Ir p/ primeira página](#)



Alocação Dinâmica - Exercício

```
void lervetor (int n)
{
    int *cont = idades;
    for (int i = 0; i < n; i++)
    {
        printf ("\nDigite o elemento %d ", i + 1);
        scanf ("%d", cont++);
    }

    void mostrarvetor (int n)
    {
        int *cont = idades;
        for (int i = 0; i < n; i++)
        {
            printf ("\nO elemento %d eh %d ", i + 1, *cont++);
        }
    }
}
```

02/02/2007

14

[Ir p/ primeira página](#)



Alocação Dinâmica - Exercício

```
main()
{
    int n;
    printf ("\n Quantas idades deseja cadastrar ? ");
    scanf ("%d",&n);
    if (criarvetor(n))
    {
        lervetor(n);
        mostrarvetor(n);
        getch();
        free (idades); // Liberar memória alocada
    }
}
```

02/02/2007

15

[Ir p/ primeira página](#)



Exercícios Propostos

- 1) Faça um programa para alocar espaço para colocar o nome do usuário (30 caracteres). Se conseguir alocar, leia esta string e mostre-a ao contrário.
- 2) "Para a família 8086 de processadores, a posição do heap não muda dependendo do modelo de memória utilizado." Verdadeiro ou Falso
- 3) As funções recursivas tem uma demanda maior de memória da pilha ou do heap ? Por que ?

02/02/2007

16

[Ir p/ primeira página](#)



Exercícios Propostos

- 4) A pilha pode colidir com o heap ? Se sim, qual a solução ?
- 5) Como é o modelo de Memória Segmentada do 8086 ?
- 6) Qual a consequência para o sistema operacional se utilizamos um ponteiro nulo ?
- 7) Qual a consequência para o sistema operacional em um mal uso da função free () ?
- 8) Qual a utilidade da função realloc () ?

02/02/2007

17

[Ir p/ primeira página](#)



Exercícios Propostos

- 9) O que significa size_t ?
- 10) Em qual biblioteca (.h) fica o sizeof ?
- 11) Qual a utilidade do sizeof no tocante a portabilidade ?

02/02/2007

18

[Ir p/ primeira página](#)

