



**USP - ICMC - SSC**  
**SSC 0501 - 1o. Semestre 2010**

**Disciplina de**  
**Introdução à Ciência da Computação**  
**ICC 1 - Teoria**

Prof. Fernando Santos Osório

Email: fosorio [at] { icmc. usp. br , gmail. com }

Página Pessoal: <http://www.icmc.usp.br/~fosorio/>

Web - WIKI ICMC: <http://wiki.icmc.usp.br/index.php/SSC-501>

PAE: Gustavo Pessin (Dout. CCMC) / <Http://pessin.googlepages.com/>

Email: pessin [at] { icmc.usp.br , gmail.com }

Monitor: Matheus Lin Alvarenga (EC) / <Http://matheuslin.wordpress.com/>

Email: matheus.lin [at] gmail.com

*Aula 10t*

**Linguagem de Programação "C"**

**Agenda:**

**REVISÃO RÁPIDA**

- **Endereços e Ponteiros**
  - Declarando e usando ponteiros e endereços
  - Alocação dinâmica de Vetores: Calloc, Malloc
  - Uso de Vetores com alocação dinâmica

**EXERCÍCIOS**

**Informações Complementares e Atualizadas:**

**Consulte REGULARMENTE**

**O material disponível na COTEIA**

### Endereços e Ponteiros:

#### Declarando Ponteiros:

```
int    *Ponteiro_Int;        /* Ponteiro para um inteiro */
double *Ponteiro_Double;    /* Ponteiro para um double */
FILE   *Arquivo;           /* Ponteiro para arquivo */
t_reg_funcionario *Ptr_Func; /* Ponteiro para um registro */
/* Criado com typedef/struct */
```

#### Usando Ponteiros:

- &** Obtém o endereço de uma variável. Ponteiros são “tipados”.  
Exemplos: `Ponteiro_Int = &Variavel_Int;`  
`Ponteiro_Double = Vetor_Double; /* ou &(Vetor_Double[0]) */`
- \*** Acessa o conteúdo de um endereço qualquer (ler ou escrever no end.)  
`*Ponteiro_Int = 10; /* Escreve o valor 10 no endereço de Ponteiro_Int */`  
`Variavel_Int = *Ponteiro_Int; /* Lê o valor contido no endereço e atribui o valor lido para Variavel_Int */`

### • Alocação Dinâmica de memória:

**calloc** - Aloca memória, zerando os dados  
**malloc** - Aloca memória, sem inicializar os dados  
**free** - Libera um bloco de memória alocada previamente

```
void *calloc ( <quantidade_elementos>, <tamanho_elemento> )
```

```
Exemplo: tabela_inteiros = calloc( 10, sizeof( int ) );
```

```
void *malloc( <quantidade_elementos> )
```

```
Exemplo: tabela_inteiros = malloc( 10 * sizeof( int ) );
```

```
void free ( void *ponteiro )
```

```
Exemplo: free ( tabela_inteiros );
```

## Vetores: Ponteiros e Endereços

### Exemplos:

```
int *ptr_int;  
double *ptr_pf;
```

```
ptr_int = (int *) calloc ( 10 , sizeof (int) );      /* Aloca 10 inteiros em seqüência */  
ptr_pf = (double *) calloc (10, sizeof (double) ); /* Aloca 10 nros. tipo double */  
free (ptr_int);                                     /* Libera a área de memória alocada */
```

### Fazendo a criação de um vetor de duas formas equivalentes:

```
int tabela[10]; /* Aloca memória: tabela aponta para o início do vetor */  
ou  
int *tabela; tabela = (int *) calloc (10, sizeof(int));
```

5

Maio 2010

## Vetores: Ponteiros e Endereços

### Exemplo: Tabela[10]

```
#include <stdio.h>  
#include <conio.h>  
  
double Tabela[10];  
  
main ()  
{  
    int i;  
  
    for (i=0; i < 10; i++)  
    {  
        printf("Dado %d = ",i);  
        scanf ("%lf",&(Tabela[i]));  
    }  
  
    printf("\nDados Lidos:\n");  
    for (i=0; i < 10; i++)  
        printf("Dado %d = %.2lf\n",i, Tabela[i]);  
  
    getch();  
}
```

### Exemplo: \*Tabela

```
#include <stdio.h>  
#include <conio.h>  
#include <stdlib.h> /* Inclui a biblioteca do "calloc" */  
  
double *Tabela; /* Cria somente o Ponteiro */  
/* Sem alocar memória */  
  
main ()  
{  
    int i;  
  
    Tabela=(double *)calloc(10,sizeof(double));  
  
    for (i=0; i < 10; i++)  
    {  
        printf("Dado %d = ",i);  
        scanf ("%lf",&(Tabela[i]));  
    }  
  
    printf("\nDados Lidos:\n");  
    for (i=0; i < 10; i++)  
        printf("Dado %d = %.2lf\n",i, Tabela[i]);  
  
    getch();  
}
```

6

Maio 2010

### Exercício:

Faça um programa para **ler dados** do disco, **adicionar dados** e depois **salvar** os dados que descrevem o **cadastro** de produtos de uma loja, contendo as seguintes informações: **código do produto** (até 5 dígitos), **nome/descrição do produto** (79 caracteres+'\0'), e **preço do produto** (em reais). O programa deve declarar um **registro** (*typedef/struct*) para agrupar os dados, que serão **lidos do disco** (arquivo "entrada.txt"), **armazenados em memória** em um **vetor com alocação dinâmica** e depois **gravados em disco** (arquivo "saida.txt").

O arquivo em disco é do tipo texto, com um dado em cada linha, onde a primeira linha do arquivo indica quantos registros estão gravados nele.

Uma vez lido o arquivo, perguntar ao usuário quantos dados ele deseja adicionar ao cadastro, criar um vetor com alocação dinâmica, e armazenar os dados digitados neste vetor. Por fim, salvar em disco os dados lidos do arquivo juntamente com os novos dados digitados pelo usuário. O novo arquivo também deve iniciar com uma primeira linha indicando o total de produtos cadastrados (os antigos mais os novos), sendo gravado em formato texto com um dado em cada linha.

```
>> Cadastro de Produtos da Loja ACME <<
** Lendo Arquivo **
Total de registros no arquivo: 30

** Entrada de Dados **

Quantos registros novos voce deseja criar? 10

Produto 1 -Codigo: 12345
Produto 1 -Descricao: Caixa_de_Bits_de_Paridade_Avulsos_(10_unidades)
Produto 1 -Preço: 1.99

Produto 2 -Codigo: 98765
Produto 2 -Descricao: Memória_WROM_Quântico-Neural_de_100_Terabytes
Produto 2 -Preço: 100.00
...
Produto 10 -Codigo: 10101
Produto 10 -Descricao: Monitor_Holístico-Holográfico_1-D
Produto 10 -Preço: 99.99

** Gravando Dados **
FIM!
```

Exemplo



**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/ssc/>**

**Página pessoal: <http://www.icmc.usp.br/~fosorio/>**

**E-mail: [fosorio \[at\] icmc. usp. br](mailto:fosorio@icmc.usp.br) ou [fosorio \[at\] gmail. com](mailto:fosorio@gmail.com)**

**PAE Gustavo Pessin – E-mail: [pessin \[at\] gmail .com](mailto:pessin@gmail.com)**

**Monitor Matheus Lin – E-mail: [matheus.lin \[at\] gmail.com](mailto:matheus.lin@gmail.com)**

**Disciplina de Introdução a Ciência da Computação**

**Web disciplina: Wiki ICMC - [Http://wiki.icmc.usp.br](http://wiki.icmc.usp.br)**

**> Programa, Material de Aulas, Critérios de Avaliação,**

**> Trabalhos Práticos, Datas das Provas, Notas**