CEFET – RJ / *Campus* Maria da Graça
Centro Federal de Educação Tecnológica
Celso Suckow da Fonseca – Rio de Janeiro

Prof. Cristiano Fuschilo

cristiano.fuschilo@cefet-rj.br

Estruturas de Dados

Bacharelado em Sistemas de Informação

COORDENAÇÃO DE Automação Industrial
Ensino de Qualidade

Estruturas de Dados

Estrutura de Dados

- Muitas vezes precisamos compor os dados para formar estruturas de dados complexas
- Variáveis compostas homogêneas (Arrays)
  - Conjunto de variáveis de mesmo tipo
- Variáveis compostas heterogêneas
  - Conjunto de variáveis de tipos diferentes
- Chamadas de:
  - Estruturas (Struct)

Struct

- Em Linguagem C Registros são chamados de Estruturas e a palavra-chave é Struct.
- Registros é um conceito de programação que é implementado pelos compiladores de cada linguagem de programação.

Aplicação de Struct

- Estruturas podem ser usadas para armazenar informações relacionadas
- Exemplo 1: Produto

Livro (char[11])	L	i	n	g	u	a	g	e	m		C
Preco (float)	59,9000										
Autor (char[11])	D	.		R	i	t	c	h	i	e	

Exemplo 2: Ficha de cliente (cadastro)

Nome (char[10])	H	e	l	e	n	a				
Idade (int)	30									
Telefone (int)	5555-5555									
Cidade (char[10])	S	a	o		P	a	u	l	o	

Definindo uma Struct

- Uma estrutura pode ser definida de formas diferentes.
- No corpo da estrutura encontram-se os membros, ou seja, as variáveis de diversos tipos que comporão esse tipo de dado heterogêneo definido pelo usuário.
- Depois de definida uma estrutura, uma (ou mais) variável do tipo estrutura deve ser definida, para permitir a manipulação dos membros da estrutura.

Forma Geral

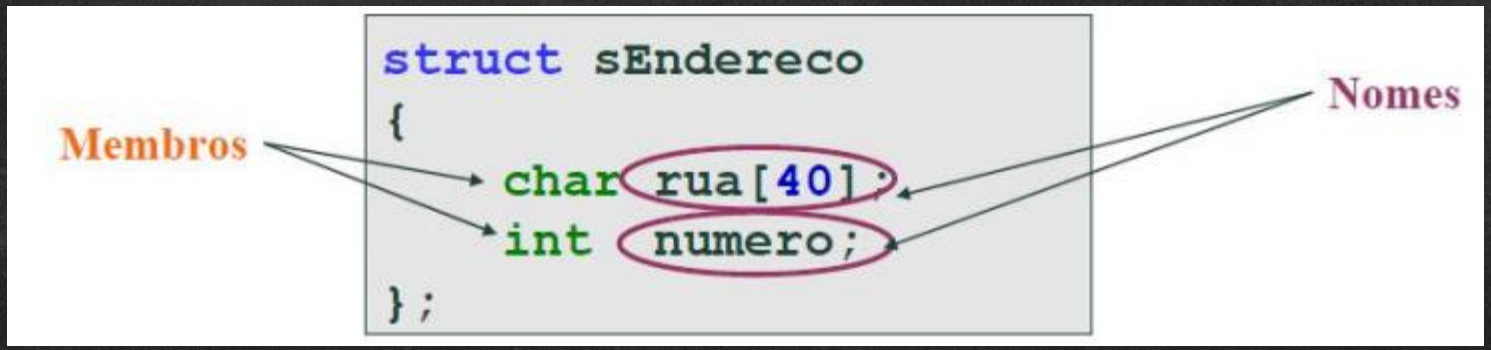
- A forma geral de declaração da Estrutura é:

```
1 struct etiqueta{
2     tipo var1;
3     tipo var2;
4     .
5     .
6     .
7     tipo varN;
8 };
```

- Onde:
  - o nome, ou **Etiqueta** (**TAG**), da estrutura é colocado logo em seguida da palavra-chave **struct**.
  - Dentro da estrutura são definidos os tipos e os campos que a compõe!

Struct

- Variáveis compostas heterogêneas (estruturas) são um conjunto de variáveis de tipos diferentes que são logicamente relacionadas.
- Essas variáveis compartilham o mesmo identificador e ocupam posições consecutivas de memória.
- Para as variáveis de uma estrutura:
  - Elas são denominadas membros;
  - São identificadas por nomes.



Exemplo

- Nesse exemplo a Etiqueta da estrutura é nomeada como ALUNO.

```
1 struct aluno {
2       int codigo;
3       char nome[200];
4       float nota;
5 };
```

- A estrutura **aluno** tem os campos código, nome e nota, respectivamente dos tipos inteiro, string e real.
- ATENÇÃO: ainda não existe alocado em memória nenhuma variável do tipo **aluno**, aqui informamos ao compilador que podemos, a partir de agora, declarar variáveis do tipo **aluno**!

Declarando Variáveis do Tipo aluno

- Para declarar a variável:
- Repete a palavra reservada struct e a etiqueta e depois define o nome da variável

```
1 struct aluno {
2       int codigo;
3       char nome[200];
4       float nota;
5 };
6 struct aluno aluno_especial, aluno_regular, aluno_ouvinte;
```

- Dessa forma foram declaradas três variáveis do tipo struct aluno, sendo: aluno_especial, aluno_regular e aluno_ouvinte.

Acessando os membros da variável do tipo Estrutura

- Para acessar os membros da estrutura, quando ela é diretamente referenciada, devemos utilizar o Ponto, que também é chamado de operador de seleção direta, veja:

```
1 aluno_especial.codigo
2 aluno_especial.nome
3 aluno_especial.nota
```

- Você pode atribuir valores aos membros das estruturas diretamente, e em qualquer parte do programa, conforme a seguir:

```
1 aluno_especial.codigo = 10;
2 strcpy(aluno_especial.nome, "Manoel");
3 aluno_especial.nota = 10.0;
```

Imprimindo os membros da Estrutura

- Você pode imprimir os membros da estrutura em qualquer parte do programa que desejar.

```
1 printf(" \n %d ", aluno_especial.codigo);
2 printf(" \n %s ", aluno_especial.nome);
3 printf(" \n %.2f ", aluno_especial.nota);
```

Obtendo Valores do Teclado

- Para obter dados do teclado devemos utilizar o **scanf**, tomando cuidado quando formos usar strings.
- Podemos obter dados do teclado em qualquer parte do programa.

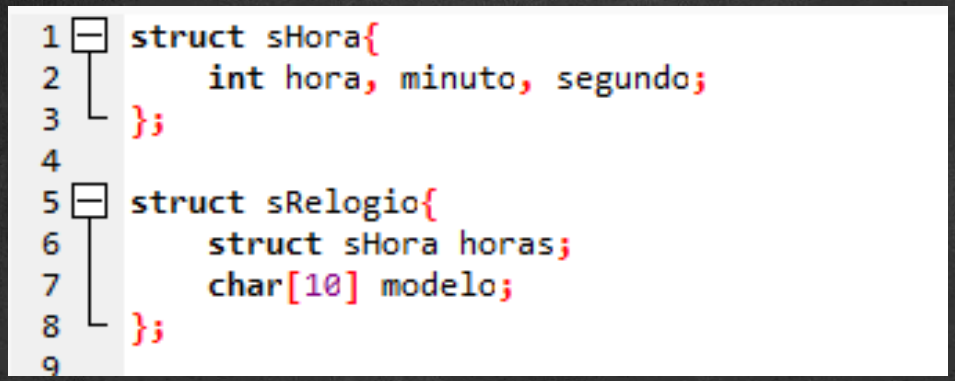
```
1 printf(" Digite o código do aluno especial: ");
2 scanf("%d%*c", &aluno_especial.codigo);
3 printf(" Digite o nome do aluno especial: ");
4 scanf("%s%*c", &aluno_especial.nome);
5 printf(" Digite a nota do aluno especial: ");
6 scanf("%f%*c", &aluno_especial.nota);
```

Estruturas Aninhadas

- Estruturas em que um ou mais de seus membros também sejam estruturas.

```
struct rotulo_1 {
        tipo1 nome1;
        tipoN nomeN;
}
struct rotulo_2 {
        struct rotulo_1 nomeX;
        tipoM nomeM;
}
```

Exemplo



```
1  struct sHora{
2      int hora, minuto, segundo;
3  };
4
5  struct sRelogio{
6      struct sHora horas;
7      char[10] modelo;
8  };
9
```

Estruturas Aninhadas

```c
#include <stdio.h>
#include <string.h>

struct sHora{
    int hora, minuto, segundo;
};

struct sRelogio{
    struct sHora horas;
    char[10] modelo;
};

int main(){
    struct sRelogio rlg;

    rlg.horas.hora = 10;
    rlg.horas.minuto = 15;
    rlg.horas.segundo = 30;

    strcpy (rlg.modelo, "Cassio");

    printf ("Modelo: %s", rlg.modelo);
    printf ("Horario: %d:%d:%d", rlg.horas.hora, rlg.horas.minuto, rlg.horas.segundo;
}

```

Vetores e Estruturas

- É possível combinar vetores e estruturas para criação de diferentes estruturas de dados.
- Podemos ter uma estrutura contendo um membro do tipo vetor, ou...
- Criar um vetor cujo os elementos sejam estruturas

Declarando vetor de Estruturas

- Dada a estrutura listada abaixo:

```
struct lista {
    char titulo[30];
    char autor[30];
    int regnum;
    double preco;
};
```

Membros do tipo vetor

- Declare um vetor com 50 elementos do tipo lista

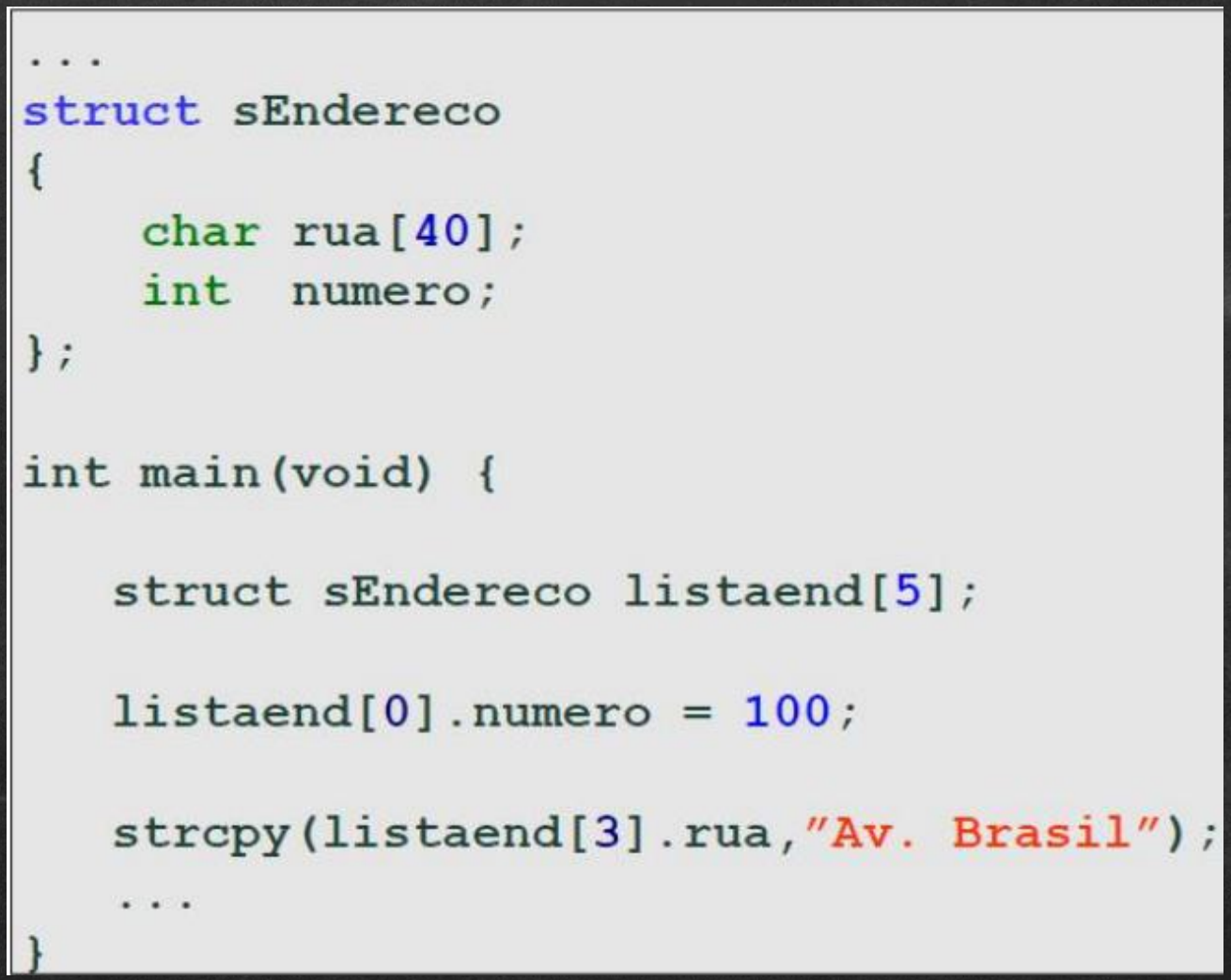
Declarando Vetores de Estruturas

struct lista livro[50];

- livro é um vetor de 50 elementos.
- Cada elemento do vetor é uma estrutura do tipo struct lista
- O que significa livro[0], livro[1], livro[2], etc?

** Por meio dessa instrução o compilador providencia espaço de memória para 50 estruturas do tipo struct lista.

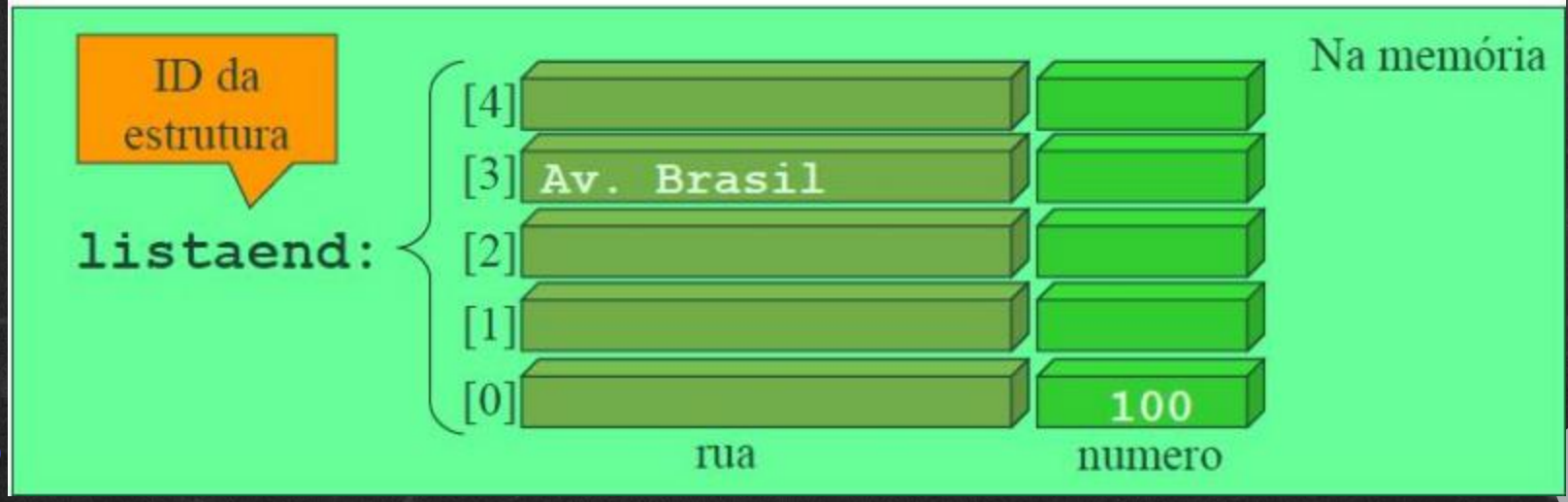
Exemplo



```
...
struct sEndereco
{
    char rua[40];
    int  numero;
};

int main(void) {

   struct sEndereco listaend[5];

   listaend[0].numero = 100;

   strcpy(listaend[3].rua,"Av. Brasil");
   ...
}
```

Trecho de exemplo

```
...
struct sEndereco
{
     char rua[40];
     int  numero;
};
int main(void) {
    struct sEndereco listaend[5];
    listaend[0].numero = 100;
    strcpy(listaend[3].rua,"Av. Brasil");
    ...
}
```



Exemplo

```c
#include <stdio.h>
struct sHora {
   int hor;
   int min;
   int seg;
};

int main(void) {

   struct sHora H[5];
   H[0].hor = 10;
   H[0].min = 15;
   H[0].seg = 30;
   printf("%d:%d:%d", H[0].hor, H[0].min, H[0].seg);
   return 0;
}
```


That's all Folks