

CCF 110 – Programação

Aula 06 – Registros (ou estruturas)

Prof. José Augusto Nacif – jnacif@ufv.br



Estrutura

- ▶ Uma estrutura agrupa várias variáveis numa só.
- ▶ Funciona como uma ficha pessoal que tenha nome, telefone e endereço. A ficha seria uma estrutura.
- ▶ A estrutura, então, serve para agrupar um conjunto de dados não similares, formando um novo tipo de dados.



Estrutura em C

- ▶ Uma estrutura é uma coleção de variáveis referenciadas por um nome, fornecendo uma maneira conveniente de se ter informações relacionadas agrupadas
- ▶ As variáveis que compreendem a estrutura são chamadas membros da estrutura (campos ou elementos).
 - ▶ Geralmente os membros são logicamente relacionados!



Sintaxe

- ▶ Para se criar uma estrutura usa-se o comando ***struct***. A palavra chave ***struct*** informa ao compilador que um modelo de estrutura está sendo definido.

- ▶ Sua forma geral é:

```
struct nome_do_tipo_da_estrutura  
{  
    tipo_1 nome_1;  
    tipo_2 nome_2;  
    ...  
    tipo_n nome_n;  
} variáveis_estrutura;
```



Sintaxe (cont.)

- ▶ **nome_do_tipo_da_estrutura** é o nome para a estrutura.
- ▶ **variáveis_estrutura** são opcionais e seriam nomes de variáveis que o usuário já estaria declarando e que seriam do tipo `nome_do_tipo_da_estrutura`.



Exemplo

```
struct ENDERECO  
{  
    char NOME[30];  
    char RUA [40];  
    char CIDADE[20];  
    char ESTADO[2];  
    int CEP;  
};
```



Variáveis

- ▶ Observe que a definição termina com um ponto-e-vírgula. Isso ocorre porque a definição de estrutura é um comando.
- ▶ Nesse ponto do código, **nenhuma variável foi de fato declarada**. Apenas a forma dos dados foi definida, ou seja, um tipo complexo de variável
- ▶ Para declarar uma variável do tipo **ENDERECO**, escreva:

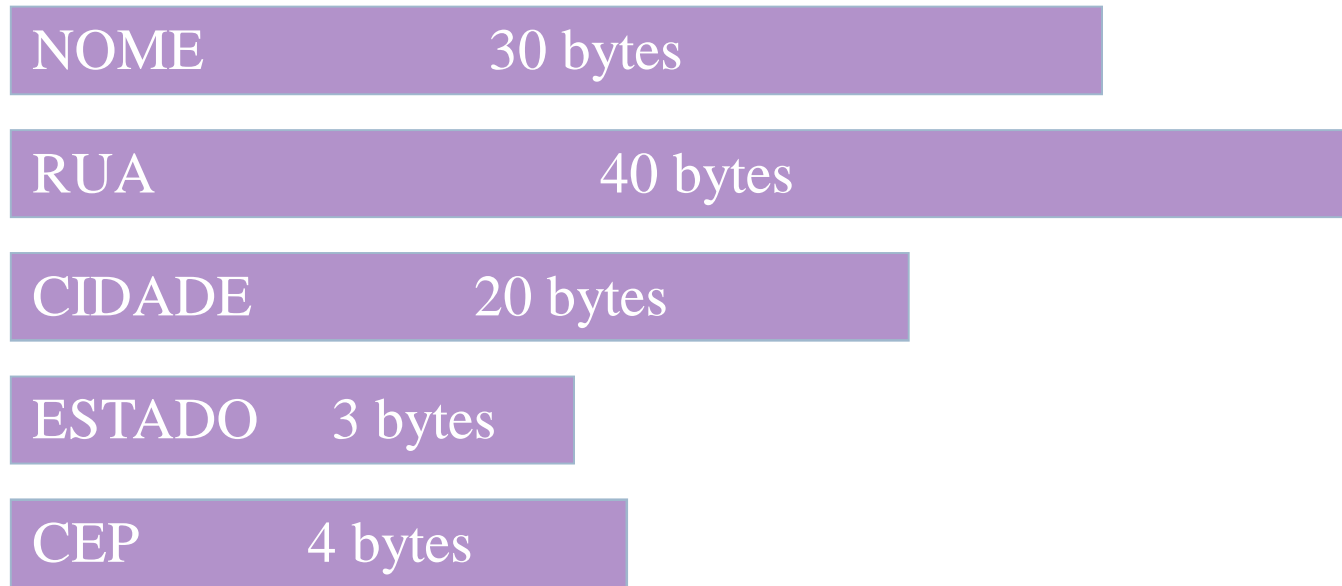
struct ENDERECO END_INFO;

Isto declara uma variável do tipo *struct endereco* chamada *end_info*



Variáveis (cont.)

- ▶ Quando uma variável de estrutura (ex: `end_info`) é declarada, o compilador aloca automaticamente memória suficiente para acomodar todos os seus membros. Como na figura:





Variáveis (cont.)

- ▶ Você também pode declarar uma ou mais variáveis logo ao definir a estrutura. Ex:

```
struct ENDERECO  
{  
    char NOME[30];  
    char RUA [40];  
    char CIDADE[20];  
    char ESTADO[2];  
    unsigned long int CEP;  
} END1_INFO, END2_INFO, END3_INFO;
```



Referenciando elementos de estrutura

- ▶ Elementos individuais de estruturas são referenciados por meio do operador ponto "."
- ▶ Ex:
 - ▶ o código seguinte atribui o CEP 12345 ao campo CEP da variável estrutura END1_INFO declarada anteriormente:

END1_INFO.CEP=12345;



Leitura e escrita

- ▶ Para imprimir o CEP na tela:

```
printf("%d", END1_INFO.CEP);
```

- ▶ Do mesmo modo, a matriz de caracteres `END1_INFO.NOME` pode ser usada para chamar `gets()`.

```
gets(END1_INFO.NOME);
```



Strings

- ▶ Da mesma forma que qualquer *string*, pode-se acessar os elementos individuais de `END1_INFO.NOME`.
 - ▶ Por exemplo, você pode imprimir o conteúdo de `END1_INFO.NOME`, um caractere por vez, usando o seguinte código:

```
int T;
```

```
gets(END1_INFO.NOME);
```

```
for(T=0; T <= strlen(END1_INFO.NOME); T++)
```

```
{
```

```
    putchar(END1_INFO.NOME[T]);
```

```
}
```

```
// ou printf("%c",END1_INFO.NOME[T]);
```



Atribuição de Estruturas

- ▶ Uma variável do tipo estrutura pode ser atribuída a outra variável do mesmo tipo estrutura, exemplo:

```
struct teste {  
    int a;  
    int b;  
    struct endereco c; //pode-se declarar dentro de outra  
                        //estrutura (acesso: x.c.CEP )  
}x,y;  
x.a=10;  
y=x; //atribui uma var estrutura a outra  
printf("%d",y.a); // irá imprimir o mesmo valor de x.a
```



Matrizes de Estruturas

- ▶ O uso mais comum de estruturas é matriz de estruturas. Para declarar uma matriz de estruturas, você deve primeiro definir uma estrutura e, então, declarar uma variável matriz desse tipo.
- ▶ Declaração de uma matriz de estruturas com 20 elementos do tipo ENDERECO, que foi definido anteriormente:

```
struct ENDERECO END_INFO[20];
```

Isso cria 20 conjuntos de variáveis que estão organizados como definido na estrutura ENDERECO.



Matrizes de Estruturas (cont.)

- ▶ Para acessar uma estrutura específica na matriz no caso unidimensional, deve-se indexar o nome da estrutura.
 - ▶ Por exemplo, para imprimir o código do CEP da estrutura 3:

```
printf("%d", END_INFO[2].CEP);
```

Como todas as outras matrizes, matrizes de estruturas começam a indexação em 0.