

Aula 10 - Estruturas

1 Objetivos

Conceituar tipos de dados definidos pelo usuário. Utilizar `struct` e `typedef` para criar registros.

2 Motivação

Dados que são relacionados podem ser salvos em uma única estrutura ou registro.

3 Aula e Exemplos

3.1 Definição

Estruturas são tipos de variáveis que agrupam dados geralmente desiguais; enquanto matrizes são tipos de variáveis que agrupam dados similares. Os itens de dados de uma estrutura são chamados **membros**, enquanto os itens de uma matriz são chamados **elementos**.

3.2 Como Declarar

Para se criar um registro, utiliza-se o comando `struct`. Sua forma geral é:

```
struct nome_do_tipo_da_estrutura {  
    tipo_1 nome_1;  
    tipo_2 nome_2;  
    tipo_3 nome_3;  
    ...  
    tipo_n nome_n;  
} var_1, var_2;
```

Veja um exemplo de registro.

```
struct ficha {  
    char nome[20];  
    int idade;  
    int RG;  
} ficha1, ficha2;
```

3.3 Como Acessar

Para se acessar um campo de um registro, digita-se o nome da variável que contém o registro, por exemplo, `ficha1` seguida de um ponto, seguida do nome do campo.

Exemplo:

```
strcpy(ficha1.nome,"João da Silva");
ficha1.idade = 5;
ficha1.RG = 567891011;
printf("Nome:%s\nidade %d\nRG %d\n",ficha1.nome,ficha1.idade, ficha1.RG);
```

Os comandos acima irão imprimir:

```
Nome:João da Silva
idade 5
RG 567891011
```

Para se acessar um elemento de um vetor:

```
ficha1.nome[3] = 'A';
```

3.4 Vetor de Registros e Registros Aninhados

Obs: O Comando `sizeof` retorna o tamanho do registro.

É possível criar um vetor de registros. Exemplo:

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

```
struct tipo_endereco{
    int numero;
    int CEP;
    char logradouro[100];
    char bairro[100];
    char cidade[100];
    char estado[3];
};
```

```
int main(void){
```

```
    struct tipo_endereco endereco[10];
```

```
    endereco[0].numero = 100;
    endereco[0].CEP = 2345678;
```

```
    strcpy(endereco[0].logradouro,"Rua Jaçana"); /*a funcao strcpy(x,y) copia o valor de y para x*/
    strcpy(endereco[0].bairro,"Perdizes");
    strcpy(endereco[0].cidade,"São Paulo");
    strcpy(endereco[0].estado,"SP");
```

```

printf("%s n.o. %d \n%s %s %s\nCEP %d\n",endereco[0].logradouro,
    endereco[0].numero,
endereco[0].bairro,
endereco[0].cidade,
endereco[0].estado ,
endereco[0].CEP);

strcpy(endereco[1].logradouro,endereco[0].logradouro);
strcpy(endereco[1].bairro,endereco[0].bairro);
strcpy(endereco[1].cidade,endereco[0].cidade);
strcpy(endereco[1].estado,endereco[0].estado);

endereco[1].numero = 200;
endereco[1].CEP = 123123;

printf("\n\n%s n.o. %d \n%s %s %s\nCEP %d\n",endereco[1].logradouro,
    endereco[1].numero,
endereco[1].bairro,
endereco[1].cidade,
endereco[1].estado ,
endereco[1].CEP);

}

```

Cuja saída será:

Rua Jaçana n.o. 100
 Perdizes São Paulo SP
 CEP 2345678

Rua Jaçana n.o. 200
 Perdizes São Paulo SP
 CEP 123123