

Vitor Amadeu Souza

vitor@cerne-tec.com.br

www.cerne-tec.com.br

Vetor de Estruturas

Na linguagem C temos os diversos tipos de dados já conhecidos, como por exemplo o char, int e long por exemplo. Agora imagine que você precise de um tipo de dado específico. Por exemplo, um tipo de dado para armazenar informações sobre uma pessoa. Neste caso, teríamos o nome da pessoa, idade, peso, endereço e sexo por exemplo. Note que a linguagem C não possui este tipo de dado específico, porém se usarmos os tipos básicos que já conhecemos em uma variável, iremos ter este tipo de dados esperado. Neste caso, chamamos este tipo de dado de registro ou estrutura. Vejamos como declarar uma estrutura no C:

```
struct nome_estrutura
{
}
}
```

No nosso caso, onde teremos na composição da variável os tipos nome, idade, peso, endereço e sexo, poderíamos definir a variável como apresentado a seguir:

```
struct funcionario
{
    char nome[50];
    char idade;
    char peso;
    char endereco[50];
    char sexo;
}
```

Para declararmos uma variável do tipo acima, usaremos a forma normal do C, onde ficará da seguinte forma a declaração de uma variável do tipo funcionário chamada func:

```
struct funcionario func;
```

No exemplo acima está sendo declarada uma variável chamada func do tipo funcionário. O acesso a cada um dos registros desta variável será feito da seguinte forma:

```
strcpy(func.nome, "Vitor");
func.idade=27;
func.peso=100;
strcpy(func.endereco, "Rio");
func.sexo='M';
```

Observe abaixo um programa C completo que funciona para aplicação apresentada acima:

```
#include <stdio.h> //Include do sistema
#include <string.h> //Include do sistema
#include <stdlib.h>
#include <fstream>

struct funcionario //Declara estrutura
{
    char nome[50];
    char idade;
    char peso;
    char endereco[50];
    char sexo;
};

main()
{
    struct funcionario func; //Declara variável do tipo estrutura

    strcpy(func.nome, "Vitor"); //Copia nome para estrutura
    func.idade=27; //Carrega idade
    func.peso=100; //Carrega peso
    strcpy(func.endereco, "Rio"); //Carrega endereço
    func.sexo='M'; //Carrega sexo

    system("pause"); //Aguarda usuário
                    //pressionar tecla do teclado
}
```

O código acima é interessante porém tem o inconveniente de suportar apenas uma informação. Neste caso podemos criar um vetor de estruturas e assim armazenar diversas informações. Vamos imaginar uma empresa com 10 funcionários, sendo assim nosso programa poderia ficar da forma abaixo por exemplo:

```
#include <stdio.h> //Include do sistema
#include <string.h> //Include do sistema
#include <stdlib.h>
#include <fstream>

struct funcionario //Declara estrutura
{
    char nome[50];
    char idade;
    char peso;
    char endereco[50];
    char sexo;
};

main()
{
    struct funcionario func[10]; //Declara vetor de estrutura

    strcpy(func[0].nome, "Vitor"); //Copia nome para estrutura
    func[0].idade=27; //Carrega idade
    func[0].peso=100; //Carrega peso
```

```
strcpy(func[0].endereco,"Rio"); //Carrega endereço
func[0].sexo='M'; //Carrega sexo

strcpy(func[1].nome,"Renata"); //Copia nome para estrutura
func[1].idade=30; //Carrega idade
func[1].peso=80; //Carrega peso
strcpy(func[1].endereco,"São Paulo"); //Carrega endereço
func[1].sexo='F'; //Carrega sexo

strcpy(func[2].nome,"Josué"); //Copia nome para estrutura
func[2].idade=20; //Carrega idade
func[2].peso=70; //Carrega peso
strcpy(func[2].endereco,"Minas"); //Carrega endereço
func[2].sexo='M'; //Carrega sexo

system("pause"); //Aguarda usuário
//pressionar tecla do teclado

}
```