Computação 2

Aula 6A

Estrutura de Dados – Registros (*structs*)

Profª. Fabiany
fabianyl@utfpr.edu.br

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Estruturas ou Registros (*structs*)

- Uma estrutura (registro) é uma coleção de variáveis referenciadas por um nome, fornecendo uma maneira conveniente de se ter informações relacionadas agrupadas.

- Uma definição de estrutura forma um modelo que pode ser usado para criar variáveis de estruturas.

- As variáveis que compreendem a estrutura são chamadas membros da estrutura (elementos ou campos).

Estruturas ou Registros (*structs*)

- Resumindo: uma estrutura ou registro é uma variável composta que contém diversas variáveis (chamadas de campos ou elementos), usualmente de tipos diferentes, mas que são logicamente relacionadas.

- Podemos comparar uma estrutura com uma ficha que possui todos os dados sobre uma determinada entidade, por exemplo:
  - Registro de alunos (Nome, curso, RA, disciplinas, médias de provas, etc...)
  - Registro de clientes (Nome, endereço, telefone, email , etc...)

Declaração de uma estrutura (struct)

- Para declarar uma estrutura utilizamos a palavra reservada **struct**, da seguinte forma:
- Sintaxe:

```
struct  identificador_da_estrutura {
    <tipo> nome_1;
    <tipo> nome_2;
    <tipo> nome_3;
    ...
    <tipo> nome_n;
};
```

< tipo > - representa qualquer um dos tipos básicos (int, float, char, double) ou tipo anteriormente definido.

Declaração de uma estrutura (struct)

- **struct** identificador_da_estrutura **{**

  <tipo> nome_1**;**

  <tipo> nome_2**;**

  <tipo> nome_3**;**

  ...

  <tipo> nome_n**;**

  };

- Observe que a definição termina com um ponto e virgula, pois a definição de uma estrutura é um comando. Além disso, o identificador(nome) da estrutura indica um especificador de tipo.

Declaração de uma estrutura (struct)

- **struct** identificador_da_estrutura **{**
  <tipo> nome_1**;**
  <tipo> nome_2**;**
  <tipo> nome_3**;**
  ...
  <tipo> nome_n**;**
  };

- Na sintaxe acima descrita, nenhuma variável foi declarada de fato, apenas a forma dos dados foi definida. Para declarar uma variável do tipo struct definido, seguimos a seguinte forma:

struct identificador_da_estrutura nomes_variáveis;

Declaração de uma estrutura (struct)

- A declaração do formato de uma estrutura pode ser feita dentro da na função *main()* ou fora dela. Usualmente, ela é feita fora da função *main*, como mostrado a seguir :

```
#include <stdio.h>
  struct   identificador_da_estrutura {
                <tipo> nome_1;
               <tipo> nome_2;
                <tipo> nome_3;
                  ...
                <tipo> nome_n;
          };

// struct identificador_da_estrutura nomes_variáveis;

void main() {
     struct identificador_da_estrutura nomes_variáveis;
      ....
}
```

Declaração de uma estrutura (struct)

- Exemplo:

```
#include <stdio.h>

  struct  pessoa{
                char nome[30], rua[50];
                int numero,idade;
   };

void main() {

     //char nome[30], rua[50];
     //int numero,idade;

    struct pessoa p;
     ....

}
```

Declaração de uma estrutura (struct)

- Também é possível declarar uma ou mais variáveis ao definir a estrutura, como mostrado a seguir:

```
#include <stdio.h>
  struct   identificador_da_estrutura {
                <tipo> nome_1;
                 <tipo> nome_2;
                <tipo> nome_3;
                    ...
                 <tipo> nome_n;
          } nomes_variáveis;

void main() {
      ....
}
```

Declaração de uma estrutura (struct)

- Exemplo:

```
#include <stdio.h>
 struct  pessoa{
                char nome[30];
                char rua[50];
                int  numero;
                int idade;
  } p1, p2;

void main() {
...
}
```

Utilizando os elementos de estruturas (structs)

- Para acessar ou modificar os elementos (campos) de uma estrutura utilizamos o operador . (ponto). O nome do identificador da estrutura seguido por um ponto e pelo nome do elemento (campo) acessa ou modifica individualmente esse elemento.

- Sintaxe

  identificador_estrutura.nome_do_campo

- Podemos usar os elementos de uma estrutura em qualquer comando que usaríamos uma variável simples.

Exemplo 1

```
#include <stdio.h>
  struct  pessoa{
                char nome[30], rua[50];
                int numero,idade;
   };

void main() {

    struct pessoa p;

    p.idade = 28;                                   //atribuição
    scanf("%i",&p.numero);                          //entrada de dados
    gets(p.nome);                                   //entrada de dados
    p.numero = p.numero  + p.idade - 100;           //expressão

    printf("O %s tem %i idade e mora na Rua %s Numero %i",
     p.nome, p.idade, p.rua, p.numero);  //saída de dados

}
```

Exemplo 2

```
#include <stdio.h>
  struct   ficha{
                    char nome[30];
                    float media;
    };

   void main() {

        struct ficha aluno1, aluno2;

         aluno1.media = 6.5
         strcpy ( aluno1.nome, “Joao Silva” );

        scanf(“%f”, &aluno2.media);
        fgets(aluno2.nome, 29, stdin);

}
```

Exemplo 3

```
#include <stdio.h>
  struct   ficha{
        char nome[30];
        int idade;
    };

   void main() {
       struct ficha aluno;

      printf("Digite o nome do aluno\n");
       scanf("%s", aluno.nome);
       printf("Digite a idade do aluno\n");
       scanf ("%i", &aluno.idade);

     printf("O aluno %s tem %i anos \n ", aluno.nome, aluno.idade);
  }
```

Exercícios

1) Crie uma *struct* chamada ponto2d que tenha como atributos os pontos x,y e z. Crie duas estruturas do tipo ponto2d chamadas ponto_inicial e ponto_final. Faça um menu com as seguintes opções e implemente-as:
   1. Digitar o valor do primeiro ponto
   2. Digitar os valores do segundo ponto
   3. Mostrar a distância entre os pontos
   4. Sair

   DICA: Distância entre dois pontos (x1,y1)(x2,y2): raiz quadrada de $(x1 - x2)^2+(y1 - y2)^2$

2) Faça um programa que realize o cadastro de quatro alunos, o cadastro de cada aluno possui nome, idade e três notas. Depois calcular a média dos alunos e mostrar quais (nome, idade, e media) estão aprovados (media >=6), a media de idade dos aprovados e quais estão reprovados (mostrar nome, idade e media).

Typedef

- Utilizado em C para redefinir um tipo de dado atribuindo-lhe um novo nome.
- Você não cria uma nova classe de dados, mas define um novo nome para um tipo já existente.
- Isso pode ser útil em casos de programas grandes, onde a alteração do tipo de uma determinada variável para outra acarretaria na alteração de muitas variáveis.
- Sintaxe

```
typedef tipo_existente novonome;
```

Typedef

- Sintaxe

  typedef tipo_existente novonome;

- Exemplo:

```
#include <stdio.h>

typedef float nota;

void main () {
      nota P1;
      printf ("Digite a nota 1\n");
      scanf ("%f", &P1);
      printf ("A nota 1 foi %f\n", P1)
}
```

Typedef

- Com o ***typedef*** é possível referenciar uma estrutura de dados dentro de outra (struct dentro de struct).

Typedef - Exemplo 1

```
typedef struct data {
    unsigned short dia;
    unsigned short mes;
    unsigned int ano;
    unsigned int idade;
} Data;

typedef struct aniversario {
    char nome[50];
    Data nascimento;
} Aniversario;

int main () {

   Aniversario Einstein, Newton;

   Einstein.nascimento.dia    = 14;
   Einstein.nascimento.mes  = 3;
   Einstein.nascimento.ano  = 1879;

   Newton.nascimento.dia    = 4;
   Newton.nascimento.mes  = 1;
   Newton.nascimento.ano   = 1643;

   Einstein.nascimento.idade = Calc_Idade ( Einstein, 2011 );
   Newton.nascimento.idade  = Calc_Idade ( Newton, 2011 );

   printf ( "A idade de Einstein seria %d \n", Einstein.nascimento.idade );
   printf ( "A idade de Newton seria %d  \n", Newton.nascimento.idade );

   getchar();
   return 0;
}
```

```
int Calc_Idade (Aniversario p, int a)
{
    int idad;

    idad = a - p.nascimento.ano;

    return idad;
}
```