3. Linguagem de Programação C

Prof. Renato Tinós

Departamento de Computação e Matemática
(FFCLRP/USP)

Principais Tópicos

3.8. Estruturas

3.8.1. Introdução

3.8.2. Uso de Estruturas

3.8.3. Vetores de Estruturas

3.8.4. Criação de Novos Tipos de Dados

3.8.5. Exercícios

3.8.1. Introdução

- **Estrutura**
  - Coleção de variáveis referenciada sobre um mesmo nome
  - Meio conveniente de manter informações relacionadas juntas
  - Conhecida também como **Registro** (Pascal) ou **Variável Composta Heterogênea**
  - Permite criar novos **tipos de dados**

3.8.1. Introdução

- **Declaração**

  – Forma uma fonte que pode ser usada para criar variáveis de estruturas

```
struct nome_da_estrutura {
        tipo nome_do_elemento_1;
        tipo nome_do_elemento_2;
        ...
};
```

3.8.1. Introdução

- **O compilador define um novo tipo de dado a partir da definição da estrutura**

- **As variáveis da estrutura são conhecidas como** elementos da estrutura **ou** membros da estrutura
  - Os elementos da estrutura são logicamente relacionados entre si

3.8.2. Uso de Estruturas

- **Para utilização da estrutura definida, variáveis do tipo de dado definido devem ser criadas**

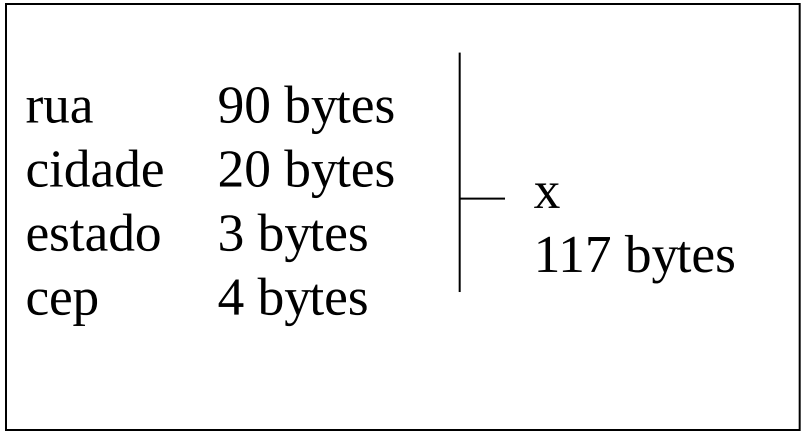
```
struct nome_da_estrutura  nome_da_variavel;
```

- **Isto declarará a variável estrutura *nome_da_variável* do tipo *nome_da_estrutura***

3.8.2. Uso de Estruturas

- **O compilador irá alocar automaticamente memória suficiente para acomodar todas as variáveis que compreendem a estrutura**
  - **Exemplo 3.8.1.** Estruturas

```
...
struct endereco {
        char rua[90];
        char cidade[20];
        char estado[3];
        unsigned long int cep;
};
...
main( ){
        struct endereco x;
...
```



3.8.2. Uso de Estruturas

- **As variáveis podem ser declaradas logo depois da estrutura**

```
struct nome_da_estrutura {
        tipo nome_do_elemento_1;
        tipo nome_do_elemento_2;
        ...
} nome_das_variaveis;
```

  - Observação: quando a estrutura é declarada fora de uma função, as variáveis neste caso são globais

3.8.2. Uso de Estruturas

- **Referência**
  - As variáveis de uma estrutura são selecionadas através do **operador ponto**

nome_da_variavel.nome_do_elemento

  - **Observação:** quando a variável é do tipo ponteiro, o operador -> é utilizado
    - Exemplo: *nome_da_variavel->nome_do_elemento*

3.8.2. Uso de Estruturas

Exemplo 3.8.2. Estruturas

```
...
struct endereco {
        char rua[50];
        char cidade[20];
        char estado[3];
        unsigned long int cep;
};
...
main( ){
        struct endereco x;
...
        strcpy(x.cidade, "Sao Paulo");
        x.cep = 14500000;
...
```

3.8.2. Uso de Estruturas

Exemplo 3.8.3. Estruturas

```
...
struct endereco {
        char rua[50];
        char cidade[20];
        char estado[3];
        unsigned long int cep;
};
...
main( ){
        struct endereco x, y;
...
        gets(y.cidade);
        printf("Cidade: %s \n",y.cidade);
...
```

3.8.2. Uso de Estruturas

Exemplo 3.8.4.
Estruturas

```
...
struct endereco {
        char rua[50];
        char cidade[20];
        char estado[3];
        unsigned long int cep;
};

struct cliente {
        char nome[100];
        struct endereco end;
        ...
};

...
main( ){
        struct cliente x;
...
        gets(x.end.cidade);
...
```

3.8.3. Vetores de Estruturas

- **Vetores e matrizes podem ser declarados como sendo do tipo de uma estrutura**

```
nome_da_estrutura  nome_da_variavel [tamanho];
```

– **Exemplo 3.8.5. :**

```
...
struct endereco {
        ...;
};
...
main( ){
        struct endereco v[25];
...
        gets(v[i].cidade);
...
```

3.8.2. Uso de Estruturas

Exemplo 3.8.6. Estruturas

```
...
struct gene{
        char nome[100];
        char organismo[100];
        double expressao[12];
        int nro_identificacao;
};
...
main( ){
        struct gene amostras[50];
        int i, j, n;
...
        for (j=0; j<50; j++)
          for (i=0; i<12; i++)
            printf("Expressao no mês %d da amostra numero %d do gene %s do organismo
%s: %f \n", i+1, amostras[j].nro_identificacao, amostras[j].nome, amostras[j].organismo,
amostras[j].expressao[i]) ;
...
```

3.8.4. Criação de Novos Tipos de Dados

- ***Typedef***
  - Permite definir novos nomes de tipos de dados
  - Forma geral

```
typedef  tipo  nome;
```

  - Exemplos:

```
...
typedef  float flutuante;
...
main ( ) {
            flutuante a;
            ...
}
```

```
...
typedef  struct {
            ...
} cliente;
...
main ( ) {
            cliente clist[N];
            ...
}
```

3.8.5. Exercícios

Exercício 3.8.1. Escrever um programa em C que

1. Possua uma estrutura chamada *aluno* com os seguintes elementos
    i. Nome (string)
    ii. Numero USP (long int)
    iii. Nota da primeira prova (float)
    iv. Nota da segunda prova (float)
    v. Média Final (float)

2. Tenha um menu que permita realizar as seguintes operações
    i. Entrar com dados dos aluno
    ii. Mostrar dados do aluno
    iii. Sair

    Observações:
    - procurar dados do aluno através de seu número USP
    - calcular a Média Final como a média aritmética entre as duas provas
    - entrar com os dados de três alunos

3.8.5. Exercícios

```
/*******************************
      Programa:Estrutura
            Ex. 3.8.1
*******************************/
#include<stdio.h>

typedef struct {
      char nome[100];
      float prova_1;
      float prova_2;
      float media;
      long int nro_usp;
} tipoaluno ;

main() {
      int i, indice_aluno;
      long int nro_usp_procurado;
      int opcao=0;
      tipoaluno aluno[3];

      while (opcao!=3) {
            // Menu
            printf("\nMenu\n");
            printf("(1) Entrar com dados do aluno\n");
            printf("(2) Mostrar dados do aluno\n");
            printf("(3) Sair\n");
            printf("Entre com a opcao: ");
            scanf("%d%*c",&opcao);
            printf("\n");
```

3.8.5. Exercícios

```
// Entrada de Dados
        if (opcao==1) {
                printf("\n Entrada de Dados\n");
                for (i=0;i<3;i++) {

                        printf("Aluno %d \n",i);
                        printf("Entre com o nome do aluno: ");
                        gets(aluno[i].nome);
                        printf("Entre com o numero USP do aluno: ");
                        scanf("%d%*c",&aluno[i].nro_usp);
                        printf("Entre com nota da primeira prova: ");
                        scanf("%f%*c",&aluno[i].prova_1);
                        printf("Entre com nota da segunda prova: ");
                        scanf("%f%*c",&aluno[i].prova_2);
                        aluno[i].media = 0.5*aluno[i].prova_1+0.5*aluno[i].prova_2;

                }
        }
```

3.8.5. Exercícios

```
        else if (opcao==2) {
                printf("\n Mostrar Dados\n");
                indice_aluno = -1;
                printf("Entre com o numero USP do aluno: ");
                scanf("%d%*c",&nro_usp_procurado);
                for (i=0;i<3;i++) {
                        if (nro_usp_procurado==aluno[i].nro_usp) {
                                indice_aluno = i;
                        }
                }
                if (indice_aluno>-1) {
                        printf("Aluno: %s \n ",aluno[indice_aluno].nome);
                        printf("Nro. Usp: %d \n ",aluno[indice_aluno].nro_usp);
                        printf("Nota 1: %f \n ",aluno[indice_aluno].prova_1);
                        printf("Nota 2: %f \n ",aluno[indice_aluno].prova_2);
                        printf("Media: %f \n ",aluno[indice_aluno].media);
                }
                else{
                        printf("Aluno nao encontrado \n");
                }

        }
  }

  printf("FIM! \n");
}
```

3.8.5. Exercícios

Exercício 3.8.2. Faça um programa, usando estruturas, que receba o instante (i.e., a hora, os minutos e os segundos) de início e término de um jogo. Então, o programa deve calcular a duração do jogo. Os valores deverão ser expressos em quantidade de horas e de minutos; e também apenas em minutos e apenas em segundos. Considere que o tempo máximo de duração de um jogo é de 24 horas e que ele pode começar em um dia e terminar em outro.