

Estrutura de Dados



Prof. Rogerio Atem de Carvalho, D. Eng.

Aula 3: Variáveis Compostas II

Structs



- Uma *struct* (ou registro) é uma variável composta heterogênea
- Constituída de uma ou mais variáveis, do mesmo ou de diferentes tipos, agrupados sequencialmente em memória e acessados por um mesmo prefixo.
- As variáveis armazenadas na estrutura são denominadas membros ou campos e devem, obviamente, possuir identificadores diferentes.
- Diferentes structs podem ter membros com o mesmo identificador

Structs



- Exemplo:
struct data
{
 unsigned dia;
 unsigned mes;
 unsigned ano;
};

Structs



- A definição da struct não declara uma nova variável e sim um novo tipo de dados que a partir daí pode ser empregado para declarar variáveis.
- Ex:
`struct data aniversario;`

Structs



- Exemplo de forma conjunta de declaração de tipo e variáveis:

```
struct data
```

```
{
```

```
    unsigned dia;
```

```
    unsigned mes;
```

```
    unsigned ano;
```

```
} aniversario, data_da_prova;
```

Structs



- A declaração da estrutura pode ser feita dentro ou fora da `main()`, se for feita fora o tipo é visível para as outras funções também.
- Uma forma de evitar o uso do termo *struct* na declaração das variáveis é empregar o termo *typedef* na declaração do tipo.

Structs



- Exemplo:

```
typedef struct data
```

```
{
```

```
    unsigned dia;
```

```
    unsigned mes;
```

```
    unsigned ano;
```

```
} tipoData;
```

```
tipoData aniversario, data_da_prova;
```

Structs



- Acessando os membros da struct:
<nome da variável>.<nome do membro>

- Exemplo:

```
printf("Digite o dia do aniversario: ");
```

```
scanf("%u", &aniversario.dia);
```

```
printf("Digite o mes do aniversario: ");
```

```
scanf("%u", &aniversario.mes);
```


Structs



- Vetores de Estruturas

```
struct nome_struct nome_vetor[tamanho_vetor];
```

- Exemplo:

```
struct data aniversarios[10];
```

ou

```
tipoData aniversarios[10];
```

- Estruturas podem ser aninhadas até a 15 níveis

Unions



- Similares às structs, porém as variáveis compartilham o mesmo espaço de memória.
- A máscara empregada depende do membro acessado.
- Depende do código-fonte específico definir como a variável será interpretada, portanto, geralmente são associados com uma variável definidora de tipo para sinalizar como interpretar o valor de uma união.

Unions



- Exemplo:

```
union valor
```

```
{
```

```
    int um_inteiro;
```

```
    float um_real;
```

```
} v;
```

v.um_inteiro será entendido com um inteiro

v.um_real será entendido como um real

Unions



- Exemplo:

```
enum tipo {inteiro, real};
```

```
union valor { int um_inteiro; float um_real; };
```

```
struct variavel
```

```
{
```

```
enum tipo tipo_da_variavel;
```

```
union valor valor_da_variavel;
```

```
}
```

Próximo Tópico



Endereçamento & Aritmética de Ponteiros