

MC 102 turma Z - 2sem2012
Linguagem C
Estruturas

Prof. Fernando Vanini
IC - Unicamp

Estruturas

- Muitas situações de programação exigem a operação com valores compostos, formados por partes.

Alguns exemplos

- *Pessoas*, representadas por
 - nome
 - endereço
 - telefone.
- *Veículos*, representados por
 - marca,
 - Modelo
 - cor
 - placa
- *Itens de estoque*, representados por
 - Código
 - Descrição
 - valor unitário
 - quantidade.

Estruturas em C

- Em C estruturas são representadas através da construção struct. Um exemplo:

```
struct{  
    char nome[30];  
    char endereco[20];  
    int idade;  
    float peso;  
} p1;
```

O exemplo declara a variável p1 como uma estrutura formada por várias partes ou campos (nome, endereco, idade e telefone).

Estruturas em C

O valor inicial de uma estrutura pode ser definido através de uma tupla. Exemplo:

```
struct{
    int ra;
    char nome[30];
    int curso;
} aluno1 = { 176761, "Jose Metralha", 42};
```

Cada valor da tupla é associado a cada campo da estrutura, pela ordem. Os tipos devem ser compatíveis um a um.

Estruturas em C

O valor de cada campo de uma estrutura pode ser atribuído individualmente. Nesse caso, o tipo valor atribuído deve ser compatível com o tipo da estrutura. Exemplos:

```
aluno1.ra = 141221;  
aluno1.curso = 34;  
...  
if(aluno.curso == x) ...  
...  
ano = aluno.ra / 10000;
```

Uma atribuição da forma `aluno1.nome = "Joaquim"`; não seria possível porque C não permite a atribuição direta de vetores (nesse caso, deve se atribuir os caracteres uma a um ou usar uma função como `strcpy()`).

Estruturas em C

Uma estrutura pode ser atribuída a outra, como um único valor. Nesse caso, as duas estruturas devem ter o mesmo tipo. Exemplo:

```
struct{
    int ra;
    char nome[30];
    int curso;
} aluno2, aluno1 = { 1777, "Jose", 42};

...
aluno2 = aluno1;
```

Os campos ra, curso e nome (!) de aluno1 são copiados para os respectivos campos de aluno2.

Estruturas em C

Uma estrutura define um novo tipo de dado. Esse tipo pode ter um nome e usado posteriormente no programa. Exemplo:

```
struct Aluno{  
    int ra;  
    char nome[30];  
    int curso;  
};
```

No exemplo, 'struct Aluno' é um novo tipo de dado, definido pelo programador. Esse novo tipo, pode ser usado para declarar outras variáveis, como mostrado abaixo.

```
struct Aluno a1, a2 = { 333, "Jose", 34 };
```

Estruturas em C

Como uma estrutura pode ser um tipo de dado, é possível construir um vetor de estruturas (de um mesmo tipo). Exemplo :

```
struct Aluno{
    int ra;
    char nome[30];
    int curso;
};

struct Aluno turma[50];
```

O exemplo define classe como um vetor de estruturas do tipo Aluno. Cada elemento desse vetor (p. ex. turma[i]) tem os campos definidos da estrutura que podem ser usados individualmente (ex. turma[i].curso).

Estruturas em C

Uma estrutura pode ser usada como parâmetro de uma função. Exemplo :

```
#define N 50
void imprime(struct Aluno a, float nota){
    if(nota >= 5.0) printf("%d => aprovado\n",a.ra)
    else printf("%d => exame\n",a.ra);
}
struct Aluno turma[N];
float nota[N];
int i;
...
for(i = 0; i < N; i++) imprime(turma[i],nota[i]);
...
```

Estruturas em C

Uma função pode retornar uma estrutura.

Exemplo :

```
#define N 50
struct Aluno turma[N]; // variável global

struct Aluno buscaPorRa(int ra) {
    int i;
    for(i = 0; i < n; i++)
        if(turma[i].ra == ra) return turma[i];
}
```