



Programação de Computadores I

Registros na Linguagem C

PROFESSORA CINTIA CAETANO

Introdução

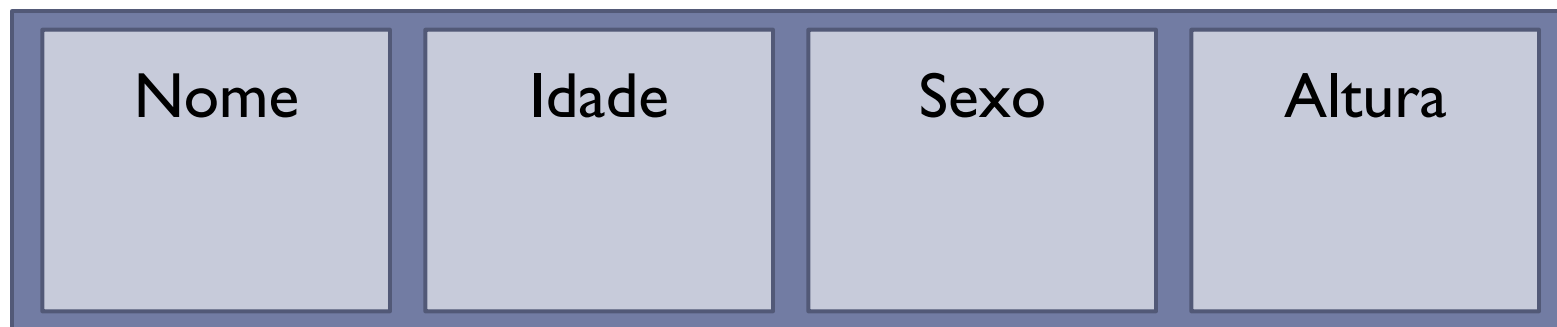
- ▶ Um vetor é capaz de armazenar diversos valores, com a restrição de que todos sejam de um mesmo tipo de dados.
- ▶ Um programa, por exemplo, que gerencie os recursos humanos de uma empresa manipula dados relativos a cada um dos funcionários que são de tipos diferentes.
- ▶ Para cada funcionário deve-se ter sua matrícula, nome, endereço, cargo, o número de dependentes, salário, data de admissão, etc.

Introdução

- ▶ A utilização de uma variável simples para cada um destes elementos, implicaria, como são vários funcionários, a necessidade de vários vetores.
- ▶ O que poderia atrapalhar a legibilidade e a manutenabilidade do programa, além de dificultar a possibilidade de armazenamento dos dados em disco.

Registro

- ▶ Um registro é uma estrutura de dados heterogênea, ou seja, formada por um conjunto de variáveis (campos) que podem assumir tipos diferentes de dados, inclusive os tipos compostos (vetores, matrizes e registros).
- ▶ Exemplo:



Declaração de Registros

- ▶ Na linguagem C, um registro é declarado através da palavra reservada **struct**.

```
struct nome_do_tipo_do_registro {  
    tipo1 campo1;  
    tipo2 campo2;  
    tipo3 campo3;  
    // ...  
    tipon campon;  
};
```

Declaração de Registros

```
struct funcionario  
{  
    char nome[30];  
    int idade;  
    char sexo;  
    float altura;  
};
```

```
void main() {  
    struct funcionario f;  
}
```

Acessando elementos

- ▶ O acesso ao campo de um registro é feito através dos identificadores do registro e do campo separados por um ponto.
- ▶ Exemplo:
 - ▶ `f.altura := 1.78;`
 - ▶ `f.sexo := 'M';`

Exemplo 1

```
#include<stdio.h>
```

```
struct funcionario
```

```
{
```

```
    char nome[30];
```

```
    int idade;
```

```
    char sexo;
```

```
    float altura;
```

```
};
```

```
void main() {
```

```
    struct funcionario f;
```

```
    strcpy(f.nome,"Joao");
```

```
    f.idade = 38;
```

```
    f.sexo='M';
```

```
    f.altura = 1.76;
```

```
    printf("Nome: %s\n", f.nome);
```

```
    printf("Idade: %d\n", f.idade);
```

```
    printf("Sexo: %c\n", f.sexo);
```

```
    printf("Altura: %4.2f\n", f.altura);
```

```
}
```

Exemplo 2

```
#include<stdio.h>
```

```
struct ficha
```

```
{
```

```
    int cod;
```

```
    char nome[40];
```

```
    char tel[10];
```

```
};
```

```
void main() {
```

```
    struct ficha aluno, func;
```

```
    printf("Código do aluno: ");
```

```
    scanf("%d", &aluno.cod);
```

```
    printf("Nome do aluno: ");
```

```
    scanf("%s", &aluno.nome);
```

```
    printf("Telefone do aluno: ");
```

```
    scanf("%s", &aluno.tel);
```

```
    printf("Código do funcionário: ");
```

```
    scanf("%d", &func.cod);
```

```
    printf("Nome do funcionário: ");
```

```
    scanf("%s", &func.nome);
```

```
    printf("Telefone do funcionário: ");
```

```
    scanf("%s", &func.tel);
```

```
    if (aluno.cod == func.cod)
```

```
        printf("Aluno e Funcionários são a  
mesma pessoa!");
```

```
    else
```

```
        printf("Aluno e Funcionários são  
pessoas diferentes!");
```

```
}
```

Registro dentro de Registro

Matr	Nome	Endereço		Cargo	Num Dep	Salário	Data Adm
		Rua	N°				
		Bairro	CEP				



Registro dentro de Registro

- ▶ Isto é possível na programação, bastando apenas que façamos a referência à variável Record mais externa, depois à mais interna e finalmente o identificador de campo.
- ▶ Caso tenhamos dois Registros que possuam campos internos iguais, é aconselhável diferenciá-los para que não haja confusão nas operações (rua1, rua2, ender1, ender2, etc).

Registro dentro de Registro

```
struct Tfuncionario
{
    char mat[8];
    char nome[30];
    struct ender
    {
        char rua[40];
        char num[6];
        char bairro[20];
        char cep[8];
    };
    char cargo[20];
    int numdep;
    float salario;
    char dataadm [8];
};
```

Exemplo 3

```
#include<stdio.h>
```

```
struct Tfuncionario
```

```
{  
    char mat[8];  
    char nome[30];  
    struct ender  
    {  
        char rua[40];  
        char num[6];  
        char bairro[20];  
        char cep[8];  
    };  
    char cargo[20];  
    int numdep;  
    float salario;  
    char dtadm[8];
```

```
};
```

```
void main(void){
```

```
    struct Tfuncionario func;  
    printf("Matricula: ");  
    gets(func.mat);  
    printf("Nome: ");  
    gets(func.nome);  
    printf("Rua: ");  
    gets(func.rua);  
    printf("Numero: ");  
    gets(func.num);  
    printf("Bairro: ");  
    gets(func.bairro);  
    printf("CEP: ");  
    gets(func.cep);  
    printf("Cargo: ");  
    gets(func.cargo);
```

Exemplo 3

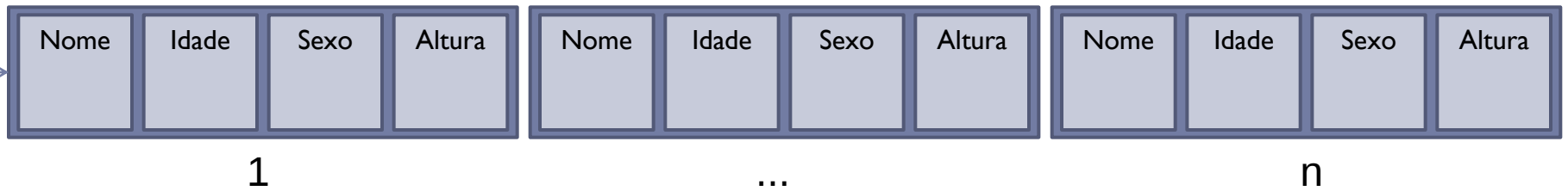
```
printf("Numero do Depto: ");  
scanf(" %d", &func.numdep);  
printf("Salário: ");  
scanf(" %f", &func.salario);  
printf("Data de Admissão: ");  
scanf(" %s", &func.dtadm);
```

```
printf("Matricula: %s\n", func.mat);  
printf("Nome: %s\n", func.nome);  
printf("Rua: %s\n", func.rua);  
printf("Numero: %s\n", func.num);  
printf("Bairro: %s\n", func.bairro);  
printf("CEP: %s\n", func.cep);  
printf("Cargo: %s\n", func.cargo);  
printf("Numero do Depto: %d\n", func.numdep);  
printf("Salário: %4.2f\n", func.salario);  
printf("Data de Admissão: %s\n", func.dtadm);
```

```
}
```

Vetor de Registros

Nome_do_Vetor



Vetor de Registros

- ▶ O uso mais comum do Struct é em vetores.
- ▶ Para declarar um vetor de struct
 - ▶ Define-se a struct;
 - ▶ Declara-se o vetor do tipo struct criado;
- ▶ Exemplo:
 - ▶ `struct aluno Turma380[28];`
 - ▶ `struct endereco vetorEndAmigos[100];`

Exemplo 4

```
#include<stdio.h>
```

```
#define MAX 5
```

```
struct contato{  
    char nome[30], tel[40];  
};
```

```
void main()  
{  
    struct contato vet[MAX];  
    int i;
```

```
    //Lendo vetor de registros  
    for(i=0;i<MAX;i++)  
    {  
        printf("Nome: ");  
        gets(vet[i].nome);
```

```
        printf("Telefone: ");
```

```
        gets(vet[i].tel);
```

```
    }
```

```
    //Imprimindo vetor de registros
```

```
    for(i=0;i<MAX;i++)
```

```
    {
```

```
        printf("\n\nNome: %s\n", vet[i].nome);
```

```
        printf("Telefone: %s\n", vet[i].tel);
```

```
    }
```

```
}
```