Aula 10: Registros

Adriano Veloso

Algoritmos e Estruturas de Dados I - DCC/UFMG

Registro/Estrutura

Muitas vezes os dados são armazenados e acessados em grupos:

- Nome do aluno
- Idade do aluno
- Nota do aluno

Registro/Estrutura

Muitas vezes os dados são armazenados e acessados em grupos:

- Nome do aluno
- Idade do aluno
- Nota do aluno

Podemos criar variáveis separadas para cada dado, e tomarmos o cuidado de processá-las conjuntamente, ou podemos acessar esses dados como uma única entidade.

Registro

- Os registros são a base para criação de estruturas complexas em C/C++. Um registro é tipo de dados composto por outros tipos de dados mais básicos.

Registro

- Os registros são a base para criação de estruturas complexas em C/C++. Um registro é tipo de dados composto por outros tipos de dados mais básicos.
- Por exemplo, podemos criar registros a partir dos tipos mais simples, tais como int, float, ou char.

Registro

- Os registros são a base para criação de estruturas complexas em C/C++. Um registro é tipo de dados composto por outros tipos de dados mais básicos.
- Por exemplo, podemos criar registros a partir dos tipos mais simples, tais como int, float, ou char.
- Mas também podemos criar registros a partir arranjos, matrizes, ou até mesmo de outros registros.

Definição

- O compilador sabe de antemão o tamanho em bytes de dados de tipo básico (i.e., int, float, ou char). No entanto o compilador não sabe de antemão o tamanho de um registro.

Definição

- O compilador sabe de antemão o tamanho em bytes de dados de tipo básico (i.e., int, float, ou char). No entanto o compilador não sabe de antemão o tamanho de um registro.
- Por isso, antes de declararmos uma variável referente ao registro, precisamos definí-lo.

Definição

- O compilador sabe de antemão o tamanho em bytes de dados de tipo básico (i.e., int, float, ou char). No entanto o compilador não sabe de antemão o tamanho de um registro.
- Por isso, antes de declararmos uma variável referente ao registro, precisamos definí-lo.
- A definição tem a seguinte sintaxe struct *nome* {*membros*};

Exemplo

```
struct aluno {
    char nome[30];
    int idade;
    float nota;
};
```

Exemplo

```
struct aluno {
    char nome[30];
    int idade;
    float nota;
};
```

Um registro pode ser composto por outros registros mais simples:

```
struct sala {
    int numero;
    aluno turma[55];
};
```

Definiç ao

- Note que para um registro fazer parte de outro registro, ele precisa ter sido definido anteriormente.

Definiç ao

- Note que para um registro fazer parte de outro registro, ele precisa ter sido definido anteriormente.
- Isso é necessário para que o compilador saiba o tamanho dos componentes do registro sendo definido.

Definiç ao

- Note que para um registro fazer parte de outro registro, ele precisa ter sido definido anteriormente.
- Isso é necessário para que o compilador saiba o tamanho dos componentes do registro sendo definido.
- A definição não aloca memória, mas apenas mostra como é a estrutura do registro, e seus componentes.

Definiç ao

- Note que para um registro fazer parte de outro registro, ele precisa ter sido definido anteriormente.
- Isso é necessário para que o compilador saiba o tamanho dos componentes do registro sendo definido.
- A definição não aloca memória, mas apenas mostra como é a estrutura do registro, e seus componentes.
- É comum dizermos que quando definimos um novo registro, criamos um novo tipo de dados.

Declaração

Para usarmos esse novo tipo de dados, precisamos declarar variáveis que sejam desse tipo.

Declaração

Para usarmos esse novo tipo de dados, precisamos declarar variáveis que sejam desse tipo.

```
int main() {
    sala aeds1;
    ...
}
```

Declaração

Para usarmos esse novo tipo de dados, precisamos declarar variáveis que sejam desse tipo.

```
int main() {
    sala aeds1;
    ...
}
```

O compilador sabe que a variável aeds1 necessita de (4+55*(30+4+8)) bytes, já que os registros foram previamente definidos.

Tamanho do registro

As vezes torna-se necessário ao programador saber o tamanho que um registro ocupa. Para isso podemos usar a função sizeof.

Tamanho do registro

As vezes torna-se necessário ao programador saber o tamanho que um registro ocupa. Para isso podemos usar a função sizeof.
Essa função precisa receber como parâmetro o registro/tipo a ser avaliado. Por exemplo, sizeof(char) irá retornar 1 byte.

Acessando um registro

- Geralmente um registro é acessado através de seus componentes.

Acessando um registro

- Geralmente um registro é acessado através de seus componentes.
- Para se acessar um componente usamos a sintaxe variável.componente.

Acessando um registro

- Geralmente um registro é acessado através de seus componentes.
- Para se acessar um componente usamos a sintaxe variável.componente.
- Por exemplo, para mudar o número da sala de aeds1: aeds1.numero=2013;

Acessando um registro

- Geralmente um registro é acessado através de seus componentes.
- Para se acessar um componente usamos a sintaxe variável.componente.
- Por exemplo, para mudar o número da sala de aeds1: aeds1.numero=2013;
- Note que quando temos registros dentro de registros, podemos precisar de mais indireções: variável.componente1.componente2.

Acessando um registro

- Geralmente um registro é acessado através de seus componentes.
- Para se acessar um componente usamos a sintaxe variável.componente.
- Por exemplo, para mudar o número da sala de aeds1: aeds1.numero=2013;
- Note que quando temos registros dentro de registros, podemos precisar de mais indireções: variável.componente1.componente2.
- Por exemplo, para mudar a nota do aluno 20 da sala aeds1: aeds1.aluno[19].nota=nova_nota;

Arranjos e funções

- Podemos declarar arranjos de registros
- ou até mesmo matrizes

Arranjos e funções

- Podemos declarar arranjos de registros
- ou até mesmo matrizes
- Um registro é passado como parâmetro da mesma forma que qualquer outro tipo (básico).

Contato

adrianov@dcc.ufmg.br