



ESTRUTURA DE DADOS

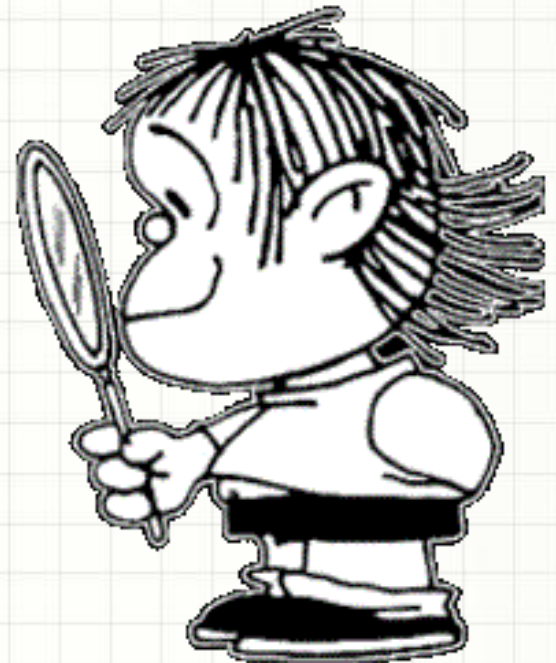
VARIÁVEIS E PONTEIROS

Prof. Dr. Daniel Caetano

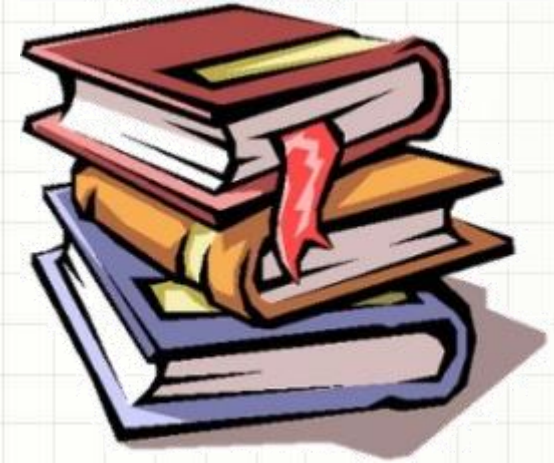
2014 - 2

Objetivos

- Compreender a organização dos dados na memória
- Compreender o que são ponteiros
- Capacitar para implementar programas com ponteiros
- **Atividade Estruturada!**



Material de Estudo



Material

Acesso ao Material

Apresentação

<http://www.caetano.eng.br/>
(Aula 11)

Material Didático

Estruturas de Dados (Parte 1) – Páginas 36 a 41



RECORDANDO...

Recordando...

- Vimos...
 - Listas Ordenadas / Não Ordenadas
 - Pilhas
 - Filas
- Armazenar dados desagregados
 - Vetor
- Armazenar dados agregados
 - Struct
- E se não sabemos o máximo de elementos?



ENDEREÇOS E PONTEIROS

Endereços e Ponteiros

- Onde fica armazenado o valor da variável?
 - Memória!
- Como é a memória do computador?
 - Um monte de “gavetas”: **posição de memória**
 - Cada gaveta tem um número único: **endereço**
- Cada nome de variável → endereço
 - Declarar variável =?
 - Armazenar valor em variável = ?

Endereços e Ponteiros

- Cada nome de variável → endereço

Endereço	0	1	2	3	4	5	6	7
Nome								
Valor	??	??	??	??	??	??	??	??

- char letra;

Endereços e Ponteiros

- Cada nome de variável → endereço

Endereço	0	1	2	3	4	5	6	7
Nome	letra							
Valor	??	??	??	??	??	??	??	??

- char letra;
- int idade;

Endereços e Ponteiros

- Cada nome de variável → endereço

Endereço	0	1	2	3	4	5	6	7
Nome	letra	idade						
Valor	??	??	??	??	??	??	??	??

- `char letra;`
- `int idade;`
- `char a[3];`

Endereços e Ponteiros

- Cada nome de variável → endereço

Endereço	0	1	2	3	4	5	6	7
Nome	letra	idade				a		
Valor	??	??	??	??	??	??	??	??

- `char letra;`
- `int idade;`
- `char a[3];`
- `letra = 'c';`

Endereços e Ponteiros

- Cada nome de variável → endereço

Endereço	0	1	2	3	4	5	6	7
Nome	letra	idade				a		
Valor	99	??	??	??	??	??	??	??

- char letra;
- int idade;
- char a[3];
- letra = 'c';
- A[1] = 'd';

Endereços e Ponteiros

- Cada nome de variável → endereço

Endereço	0	1	2	3	4	5	6	7
Nome	letra	idade				a		
Valor	99	??	??	??	??	??	100	??

- char letra;
- int idade;
- char a[3];
- letra = 'c';
- A[1] = 'd';
- idade = 256;

Endereços e Ponteiros

- Cada nome de variável → endereço

Endereço	0	1	2	3	4	5	6	7
Nome	letra	idade				a		
Valor	99	0	1	0	0	??	100	??

- char letra;
- int idade;
- char a[3];
- letra = 'c';
- A[1] = 'd';
- idade = 256;

Como saber o endereço de uma variável?

Prefixo &

Endereços e Ponteiros

- Exemplo

```
int a=1, b=2;
cout << "Val.A: " << a <<endl;
cout << "End.A: " << &a <<endl;
cout << "Val.B: " << b <<endl;
cout << "End.B: " << &b <<endl;
```

Endereços e Ponteiros

- Podemos guardar endereço em uma variável?

```
int a=1, b;  
b = &a;  
cout << "End.A: " << &a <<endl;  
cout << "Val.B: " << b <<endl;
```

- Funciona?
- Precisamos de um tipo novo de variável...



PONTEIROS

Ponteiros

- Ponteiro: serve para armazenar um endereço
 - Indicado por um * na frente do nome da variável

```
int a=1, *b;
```

```
b = &a;
```

```
cout << "End.A: " << &a <<endl;
```

```
cout << "Val.B: " << b <<endl;
```

- E se quisermos saber o valor da variável “**apontada**”?
 - Usamos o * na frente do nome do ponteiro

Ponteiros

- Exemplo: Lendo valor apontado

```
int a=1, *b;  
b = &a;  
cout << "Val.A: " << a <<endl;  
cout << "End.A: " << &a <<endl;  
cout << "Val.B: " << b <<endl;  
cout << "Val.*B: " << *b <<endl;
```

Ponteiros

- Acompanhe o exemplo:
 - Conhecendo o básico sobre ponteiros.

Ponteiros

- Acompanhe outro exemplo:
 - Mudando valores usando ponteiros.

Ponteiros

- Acompanhe outro exemplo:
 - Ponteiros nulos: NULL.

Ponteiros

- Acompanhe outro exemplo:
 - Ponteiros para estruturas.

Ponteiros

- Acompanhe outro exemplo:
 - Ponteiros para estruturas e o referenciador -> .



ALOCAÇÃO DINÂMICA DE MEMÓRIA

Alocação Dinâmica de Memória

- Alocar memória estaticamente: declarar
 - int, float, char etc.
- Alocação dinâmica?
 - Reservar no momento necessário
 - Liberar quando não for mais necessário
- Reservar: **new**
- Liberar: **delete**

Alocação Dinâmica de Memória

- Modo de Usar: **new** e **delete**

```
int *p;  
p = new int;  
*p = 10;  
cout << p << endl;  
cout << *p << endl;  
delete p;
```

- Acompanhe os exemplos

Exemplo Alocação Dinâmica

- Crie uma estrutura que represente um **produto** em estoque, contendo:
 - um número **identificador**
 - uma **quantidade** disponível
 - **preço** atual do produto.
- Crie uma função que **imprime** um produto.

Exemplo Alocação Dinâmica

- Faça um programa que crie um produto “em tempo de execução” com o **new** e, depois de imprimi-lo, remova-o com o **delete**.

Exemplo Alocação Dinâmica

- Modifique o programa anterior para que ele crie **3** produtos em tempo de execução, imprima os **3** e, posteriormente, remova-os da memória.
- Modifique o programa anterior para que, além de imprimir os conteúdos de cada produto, **também imprima o endereço na memória** de cada produto.



CONCLUSÕES

Resumo

- Além do uso normal das variáveis, podemos verificar e usar seus endereços...
 - Que serão importantes para criar estruturas de dados de tamanho indeterminado!
-
- Como usar esse conceitos para aplicativos mais úteis?



PERGUNTAS?



EXERCÍCIOS DE FIXAÇÃO

Exercício 1

- Usando o operador **&** podemos ver o endereço de uma variável. Você consegue bolar um esquema para identificar o “tamanho” (espaço na memória, em bytes) gasto por uma variável do tipo **int**?

Exercício 2

- Usando o mesmo esquema do exercício anterior, identifique o tamanho das variáveis dos tipos:
 - char
 - long
 - float
 - double

Exercício 3

- Nos exercícios da aula trabalhamos com duas estruturas: **aluno** e **produto**. Identifique o tamanho que cada elemento do tipo dessas estruturas ocupa na memória.

Exercício 4

- Compare os resultados obtidos nos exercícios anteriores com os resultados obtidos pela função “sizeof”

sizeof(int)