



(Estrutura de Dados)

Aula 6

Prof. Dr. Eliseu LS



(Linguagem C ++)
(PILHA ESTATICA)

Prof. Dr. Eliseu LS

Conceito de Ponteiro

No C/C++ toda variável ao ser declarada possui também um endereço de memória que pode ser acessado através do **caractere &**.

- Considere a declaração: **int aux = 10;** => O nome do endereço de memória de **aux** será definido automaticamente como **&aux**
- Para exibir o conteúdo de **aux** => **cout << aux ;**
- Para exibir o endereço de memória de **aux** = **cout << &aux ;**

Declaração de um Ponteiro

Ponteiro é uma variável que serve para armazenar o endereço de outra variável, exatamente do mesmo tipo. Todo ponteiro, ao ser criado obrigatório deve ser inicializado antes de ser usado.

- Criando o ponteiro *ptx

int *ptx;

- Inicializando o ponteiro *ptx, isto é, apontando ptx para um endereço de memória

ptx = &aux; ou ptx = NULL;

Programa 1 - Ponteiro apontando para uma variável

Todo ponteiro deverá ser inicializado logo depois de criado, um ponteiro têm que ser exatamente do tipo da variável que ele irá armazenar o valor ou apontar.

No programa abaixo, a variável **aux** têm um endereço de memória chamado **&aux** e um ponteiro apontando para ela que é o ***ptaux**. Para apontar um ponteiro para o endereço de uma variável, o asterisco deve ser retirado: **ptaux = &aux**

```
#include "iostream"
#include "cstdlib"
using namespace std;
int main () {  setlocale(LC_ALL, "Portuguese");
int aux = 10; // cria uma variável inteira
int *ptaux; // cria um ponteiro inteiro
ptaux = &aux; // inicializa o ponteiro
cout<< "\nO valor de aux é:" << aux;
cout<< "\nO endereço de memória de aux é :" << &aux;
cout<< "\nO endereço de memória de aux é :" << ptaux;
cout<< "\nO endereço de memória ptaux é:" << &ptaux;
cout<< "\nO valor de aux é:" << *ptaux;
system("pause"); return 0; }
```

Programa 2 - Ponteiros como argumentos em sub rotinas

Neste Swap com ponteiros, o void **trocar** () possui dois ponteiros como argumentos, isso significa que teremos que enviar dois endereços de memória um para o ponteiro ***x** e outro para o ponteiro ***y**.

A variável **aux** recebe o conteúdo do **ponteiro *x**. o **ponteiro *x** recebe o conteúdo do **ponteiro *y**, o **ponteiro *y** recebe o conteúdo de **aux** que é um número qualquer.

No main(), ao executar o void trocar () os endereços de **&a** e **&b** são enviados para dentro dos argumentos ***x** e ***y**.

```
#include "iostream"
#include "cstdlib"
using namespace std;
void trocar (int *x, int *y) { int aux;
    aux = *x;
    *x = *y;
    *y = aux; }

int main() { int a = 10,      b = 20;
    cout << "\nValor inicial de A:" << a ;
    cout << "\nValor inicial de B:" << b ; system("pause");
    trocar ( &a, &b );
    cout << "\nValor Final de A:" << a ;
    cout << "\nValor Final de B:" << b ; system("pause"); }
```

(Alias) Como um Struct se torna um Datatype

Ao criar uma estrutura do tipo struct, praticamente você cria também um novo datatype dentro do programa, por exemplo:

```
typedef struct estr1 tes1;
```

```
struct estr1 {  
    int x[10];  
};
```

O struct **estr1** recebeu um alias (apelido) de **tes1**, agora existe um novo datatype no código que é justamente **tes1**, todas as vezes que quiser utilizar esta estrutura você deve criar um objeto (variável) do tipo **tes1**.

```
tes1 m1;                                for (int i=0; i<10;i++ ) m1.x [ i ] = i;  
m1.x[0] = 1;                            for (int i=0; i<10;i++ ) cout << m1.x [ i ] << endl;  
m1.x[1] = 2;
```

Referenciando um Struct através de um Ponteiro

Considerando que agora temos um novo datatype chamado `tes1`, podemos criar um ponteiro do mesmo tipo (`tes1`) que irá apontar para um struct do tipo `tes1`.

```
tes1 m1;    // cria o objeto m1 do tipo tes1 ( um struct )
tes1 *ptm1; // cria o ponteiro *ptm1 do tipo tes1, isto é que irá receber um endereço do
tipo tes1
ptm1 = &m1; // apontamos ptm1 para m1, isto é, colocamos o endereço de m1 dentro de
ptm1
```

Ao apontarmos o ponteiro ***ptm1** para o endereço **&m1**, o ponteiro ***ptm1** se torna uma espécie de clone do struct **m1**. Agora podemos acessar todos os atributos de **m1** através do **ptm1** como se estivéssemos manipulando diretamente o **m1**. “**O ponteiro ptm1 se tornou o struct m1**”.

```
ptm1-> x[0] = 1;           for (int i=0; i<3; i++) cout << ptm1->x [ i ] << endl;
ptm1-> x[1] = 2;
```

Ao contrário do que acontece ao apontarmos um ponteiro para uma variável, para acessar o valor dos atributos usa-se o ponteiro sem o asterisco é com uma seta no lugar do ponto.

Programa 3 - Struct com Ponteiro

```
#include "iostream"
#include "cstdlib"
#define n 2
using namespace std;
typedef struct mynotas notas;
struct mynotas {
    string nome [n]; double n1[n]; double n2[n]; double media[n]; };

notas tb;
notas *ptb = &tb;
int linha = -1;

double calMedia (double * a , double * b) {
    return ( (*a + *b) / 2); }

void newLinha ( ) { double a, b; string nome;
    if ( linha == n-1 ) return;
    linha ++;
    cout << "Nota1:"; cin >> a; ptb->n1 [ linha ] = a;
    cout << "Nota2:"; cin >> b; ptb->n2 [ linha ] = b;
    cin.ignore(); cout<< "Nome :"; getline( cin, nome ); ptb->nome [ linha ] = nome;
    ptb->media[ linha ] = calMedia ( &a, &b ); }

void exibir() { system("clear");
    cout << "*** Exibição de Dados ***\n";
    for ( int i = 0; i <= linha; i++)
        cout<< ptb->nome[i] << " " << ptb->n1[i]<< " , " << ptb->n2[i]<< " media=" << ptb->media[i]<< endl;
    system("sleep 4"); }

int main () { setlocale (LC_ALL, "Portuguese");
    newLinha(); newLinha(); exibir(); return 0; }
```

TAREFA / AVALIAÇÃO CONTINUADA

I - Responda as questões a seguir

1. Defina o que seria um ponteiro. Explique e dê exemplos.
2. O que seria o endereço de uma variável. Explique como é criado e como posso exibi-lo.
3. Como posso acessar o conteúdo de uma variável usando um ponteiro. Explique e dê exemplo.
4. Como posso acessar o endereço de uma variável usando ponteiro. Explique e dê exemplo.
5. Qual seria o tipo de ponteiro para a STRUCT PESSOA?
6. Declare duas variáveis e dois ponteiros. Exiba o resultado da soma dessas variáveis somando os seus ponteiros (com Asterisco).

II - Faça um programa com menu de três opções (1-ler linha, 2- mostrar e 3- sair), contendo um STRUCT para cadastrar o nome, a idade, o sexo, salário e salarioNovo de pelo menos duas pessoas. Crie um ponteiro para manipular o STRUCT. O programa deverá conter as seguintes sub rotinas abaixo: (use switch case)

- a) void NovaLinhaStruct ()
- b) void listarLinhasStruct ()
- c) ler_idade ()
- d) ler_sexo ()
- e) ler_nome ()
- f) ler_salario ()
- g) getAumento (double * salario) // aumento de 10% sobre o salário digitado

NOTA: Entregar o código fonte e o print de execução dentro de um único arquivo DOCX ou PDF, somente através do Teams.