

Ejemplo C++ Punteros

Michael Daniel Murillo López

ID:534830

Estructura de Datos

Corporación Universitaria Minuto de Dios

Uniminuto

Bogota D,C

2 de octubre de 2017

1. Código:

El siguiente documento nos presentará el paso a paso de un ejemplo que se realizo en c++ donde manejamos los punteros en dos tipos que son:

- * Ir a la dirección de memoria
- * La posición.

Explicaremos cada línea de código en un pequeño resumen también mostrando el pantallazo de nuestro programa para que quien lo lea nos entienda mejor.

```

#include <iostream>

using namespace std;

int main()
{
    int a;
    int b;

    cout << "Ingresa el valor de a: ";
    cin >> a;
    cout << endl;

    cout << "Ingresa el valor de b: ";
    cin >> b;
    cout << endl;

    // Punteros de tipo entero
    int *p;
    int *p2;

    cout << "Antes" << endl;
    cout << "Variable a = " << a << endl;

    cout << "Direccion de a = " << &a << endl << endl;

    cout << "Variable b = " << b << endl;
    cout << "Direccion de b = " << &b << endl << endl;

    // Tiene basura al principio, podria inicializar con *p=0
    cout << "Contenido de p (Basura)= " << *p << endl;
    cout << "Direccion de p = " << &p << endl << endl;

    cout << "Despues" << endl;
    a++;
    p = &a;

    cout << "Contenido de p = " << *p << endl;
    // p ahora tiene la dirección de b
    p = &b;
    // Le sumo 20 al contenido de p, es decir, estoy incrementando el valor de b
    *p +=20;

```

En primer lugar creamos la clase main() y declaramos dos variables a y b de tipo entero (int) y luego cargamos cada una pidiendo que el usuario ingrese el valor de las mismas, luego de esto creamos los punteros para las variables nombradas anteriormente e imprimimos el valor que tiene cargado cada variable y su dirección en memoria, así mismo también nos trae el contenido que tiene la variable de ese puntero. Luego a la variable 'a' le sumamos 1 e imprimimos el puntero 'p' con su dirección en memoria. Luego de esto a 'p' se le da la dirección de 'b' y se le suma 20.

Figura 1: Captura 1

```

cout << "Variable a = " << a << endl;
cout << "Variable b = " << b << endl << endl;

// p ahora tiene la dirección de a
p=&a;

// Contenido de p es igual al contenido de a * 5
*p = a * 5;

cout << "Contenido de p = " << *p << endl;
cout << "Variable a = " << a << endl << endl;

// Tiene basura al principio, podría inicializar con *p2=0
cout << "Contenido de p2 (Basura) = " << *p2 << endl;
cout << "Dirección de p2 = " << &p2 << endl << endl;

// El contenido de p es asignado al contenido de p2
p2 = p;

// Incremento 15 al contenido de p2
*p2 += 15;

cout << "Contenido de p2 = " << *p2 << endl;
// p apunta a otra dirección de memoria, se desplaza 4 bytes en memoria
p++;

// El contenido de esa nueva dirección
cout << "Contenido de p (Basura) = " << *p << endl;

cout << endl;
cout << "----- MATRICES Y ARITMÉTICA DE PUNTEROS ----- " << endl;

// Puntero a una matriz
int **pm;

int cols;
int rows;

cout << "Ingresa nro de filas: ";
cin >> rows;

```

Aca imprimimos la variable 'a' y 'b' con los cambios y sumatorias hechos anteriormente, a 'p' se le da la dirección de memoria de la variable 'a' para luego a esta misma multiplicarla por 5, luego imprimimos la dirección del puntero 'p' y el contenido de ese mismo puntero. El segundo puntero 'p2' lo igualamos al primer puntero 'p' y le incrementamos 15 para luego imprimir el contenido de 'p2'. Luego de esto a 'p' se le suma 1 y se imprime su dirección de memoria y finalizamos estas matrices. Inmediatamente comenzamos con las Matrices y Aritmética de punteros, donde declaramos un puntero a otro puntero de una matriz 'pm', también creamos dos variables 'cols' y 'rows' como enteros (int) y le pedimos al usuario el número de filas que desea.

Figura 2: Captura 2

```

cout << endl;
cout << "Ingresa nro de columnas: ";
cin >> cols;

pm = new int* [rows];
for (int i = 0; i < rows; i++) {
    pm[i] = new int[cols];
}

cout << "Elementos de la Matriz con sus direcciones: " << endl;
for (int i = 0; i < rows; i++) {
    for (int j = 0; j < cols; j++) {
        pm[i][j] = i + j;
        cout << pm[i][j] << "--> ";
        cout << &pm[i][j] << " ";
    }
    cout << endl;
}
cout << endl;

cout << "Elementos de la Matriz con sus direcciones, con aritmética de punteros: " << endl;
for (int i = 0; i < rows; i++) {
    for (int j = 0; j < cols; j++) {
        // Aritmética de punteros
        (*(pm + i) + j) = i + j;
        cout << (*(pm + i) + j) << "--> ";
        cout << &pm[i][j] << " ";
    }
    cout << endl;
}

// Elimino cada vector de la matriz
for (int i = 0; i < rows; i++) {
    delete[] pm[i];
}

// Elimino el vector principal
delete[] pm;

return 0;
}

```

En esta empezamos pidiendo al usuario el número de columnas que desea. Luego de esto igualamos la matriz 'pm' al puntero de 'rows' para luego iterar por medio de un ciclo for entre filas y columnas de la matriz 'pm'. Luego de esto imprimimos a través de un ciclo for anidado los elementos de la matriz con sus respectivos contenidos de cada posición de la matriz. Luego de esto y también gracias a un ciclo for anidado imprimimos los elementos de la matriz con sus direcciones pero con la aritmética de punteros y al final eliminamos cada vector de la matriz con el vector principal de 'pm' y retornamos 0.

Figura 3: Captura 3