**Aritmética de Apuntadores**

Se pueden realizar varias operaciones con los apuntadores, por ejemplo:

* Incrementar un apuntador (++)
* Decrementar un apuntador (--)
* Sumar un entero a un apuntador (+0+=)
* Restar un entero a un apuntador (-0-=)
* Restar un apuntador de otro apuntador del mismo tipo

Operaciones no validas:

* No se pueden sumar dos punteros
* No se pueden multiplicar dos punteros
* No se pueden dividir dos punteros

**EJEMPLO:**

Se tiene declarado el arreglo int v[5], y su primer elemento se encuentra en la ubicación de memoria 3000.

Se ha inicializado el apuntador vPtr para que apunte a v[0], (es decir, el valor de Ptr es 3000).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ubicación | 3000 | 3004 | 3008 | 3012 | 3016 |
| Variable apuntador  vPtr | V [0] | V [1] | V [2] | V [3] | V [4] |

Suponiendo que un int se almacena en 4 bytes de memoria.

vPtr se puede inicializar para que apunte al arreglo v como se muestra a continuación:

* int \*vPtr = v;
* int \*vPtr = &v [0];

Debido a que el nombre de un arreglo es equivalente a la dirección de su primer elemento.

En la aritmética convencional, la suma 3000+2 produce el valor 3002. Por lo general este no es el caso de la aritmética de apuntadores.

Cuando se suma (o resta) un entero a un apuntador, no se incrementa o decrementa simplemente debido a ese entero, sino por ese entero multiplicado por el tamaño del objeto al que el apuntador hace referencia.

El numero de bytes depende del tipo de datos del objeto. Por ejemplo, vPtr + 2 produciría 3008(3000 2\*4).

Suponiendo que un int se almacena en 4 bytes de memoria. En el arreglo v, vPtr apuntaría ahora a v [2].

Si se almacena un entero en 2 bytes de memoria, se tendría la ubicación de memoria 3004 (3000 + 2\*2).

Los apuntadores se pueden comparar mediante el uso de los operadores de igualdad y relacionales.

Los operadores relacionales se utilizan con apuntadores que apuntan a miembros del mismo arreglo.

Las comparaciones con apuntadores comparan las direcciones almacenadas en los apuntadores.

Una comparación de dos apuntadores que apuntan al mismo arreglo podría mostrar, por ejemplo, que un apuntador apunta a un elemento de mayor numeración del arreglo que el otro apuntador.

Un uso común de la comparación de apuntadores es determinar si un apuntador es 0 (es decir, si el apuntador es nulo; que no apunta a nada).

**Apuntadores y arreglos**

Los apuntadores se pueden utilizar para realizar cualquier operación en la que se involucren los subíndices de arreglos.

Si se tienen las siguientes declaraciones:

* Int b [5]; // Crea el arreglo int b de 5 elementos
* Int \*bPtr; // Crea el apuntador int bPtr

Es posible establecer **bPtr** como la dirección del primer elemento en el arreglo b con la instrucción:

* bPtr = b; // asigna la dirección del arreglo b a bPtr

ya que el nombre del arreglo (sin subíndice) es un apuntador (constante) al primer elemento del arreglo.

Esta instrucción es equivalente a asignar la dirección del primer elemento del arreglo:

* bPtr = &b [0]; // también asigna la dirección del arreglo b a bPtr

el elemento b [3] del arreglo se puede referenciar con la siguiente expresión de apuntador:

* (bPtr + 3)

Cuando el apuntador apunta al inicio de un arreglo, el desplazamiento indica a cuál elemento del arreglo se debe hacer referencia, y el valor del desplazamiento es idéntico al subíndice.

La notación anterior se llama notación apuntador/desplazamiento.

Los paréntesis son necesarios, ya que la precedencia de \* es mayor que la precedencia de +.

Sin los paréntesis, la expresión sumaria 3 al valor de \*bPtr.

En cuanto a las direcciones de apuntadores, las siguientes expresiones son equivalentes:

* &b [3] es igual bPtr + 3

El nombre del arreglo se puede tratar como apuntador y se puede utilizar en la aritmética de apuntadores.

**Punteros a cadenas**

Si se tiene el siguiente array de caracteres:

char alfabeto [27] = “ABCDEFGHIJKLMNOPQRSTUVWXYZ”;

Declaramos ahora **p** un puntero de tipo **char** a alfabeto;

char \*p;

Se establece que **p** apunta al primer carácter de alfabeto

p = alfabeto;

p = &alfabeto [0] es equivalente a p = alfabeto

Escribimos la sentencia cout<<\*p<<endl;

Se visualiza la letra A, ya que p apunta al primer elemento de la cadena.

* Formas de definiciones de cadenas:

Char cadena1 [] =” Hola mundo”; Array que contiene una cadena

Char \*cptr = “C++ a su alcance”; Puntero a cadena, el sistema reserva memoria para la cadena.

Al realizar aritmética de apuntadores con un arreglo de caracteres, los resultados serán consistentes con la aritmética regular, ya que cada carácter tiene un tamaño de 1 byte.

Ejercicio 1: implementar un método o función que cuente el numero de caracteres de una cadena mediante apuntadores.

Ejercicio 2: implementar un método o una función que tome una cadena de caracteres como entrada y la imprima al revés utilizando apuntadores.