Lenguaje de programación C++

Módulo 6: Tópicos Avanzados

1. Conceptos Básicos de Apuntadores

Apuntador

La computadora guarda las variables en la memoria y cada espacio de memoria tiene una dirección. Normalmente no necesitamos saber la dirección en la que está guardada una variable, ya que nos referimos a ella por su nombre.

En C++ existen variables que puede guardar una dirección de memoria, se llaman apuntadores y una variable de tipo apuntador puede guardar direcciones de variables de un tipo determinado; es decir, hay apuntadores a entero, apuntadores a double, etc.

Para declarar un apuntador se utiliza el siguiente formato:

```
tipo *nombreVariable;
```

Por ejemplo:

```
int *ptr1;      // apuntador a entero
double *ptr2; // apuntador a double
char *ptr3;    // apuntador a caracter
```

Nota que el * puede ir pegado al tipo de dato o a la variable.

Si escribes la siguiente expresión en un programa de C++:

```
ptr1 = ptr2;
```

el compilador marca un error porque ptr1 y ptr2 son apuntadores de diferente tipo.

En la siguiente declaración:

```
int *p, n;
```

p es una variable de tipo apuntador a entero, mientras que n es una variable de tipo entero.

Operador dirección &

El operador & se utiliza para obtener la dirección de una variable y se puede usar con el siguiente formato:

```
&nombreVar;
```

Por ejemplo:

```
int *p, n;     // declaro p como un apuntador a entero y n como un entero
p = &n;       // asigno la dirección de la variable n en la variable p
```

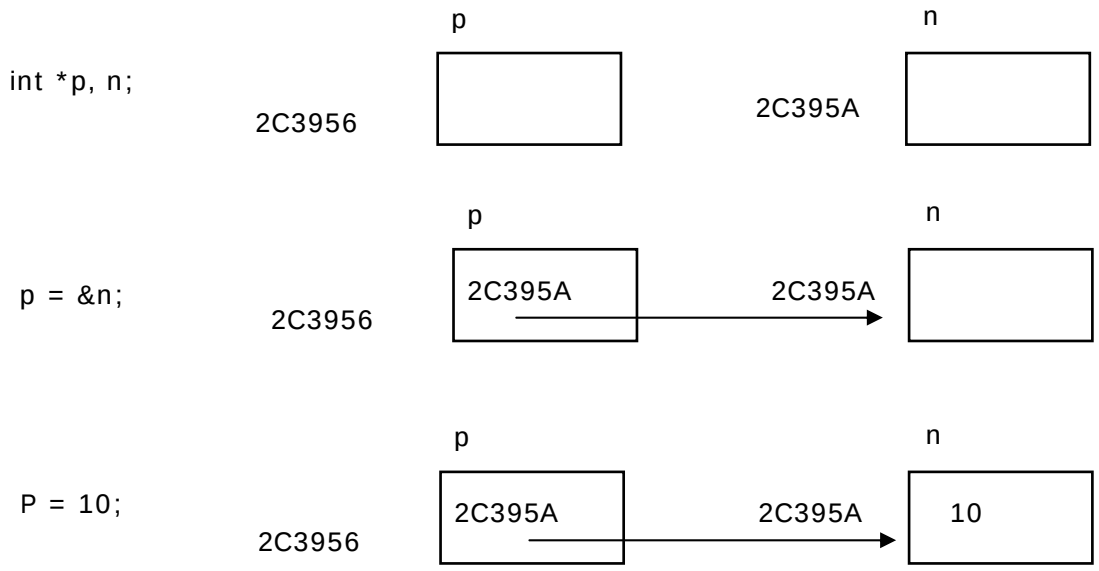
Operador *

El operador * aplicado a una variable de tipo apuntador genera la variable a la cual apunta, por ejemplo:

```
int *p, n;
p = &n;
```

En este ejemplo la expresión ***p** se refiere a la variable n.

Veamos ahora este ejemplo gráfico:



El valor NULL (nulo)

Para indicar que una variable no apunta a nada se utiliza el valor Nulo (NULL) como se muestra en el siguiente ejemplo:

```
int *p;
p = NULL;
```

Desplegar un apuntador

Es posible mostrar el contenido de una variable de tipo apuntador en la pantalla utilizando un cout como se muestra en el siguiente ejemplo:

```
int main()
{
  int x, y, *p, *q;
```

```
  p = &x;
  *p = 5;
  q = &y;
  *q = 23;
  cout<<*p<<" "<<*q<<endl;
  q = p;
  *p = 35;
  cout<<*p<<" "<<*q<<endl;
  q = NULL;
  cout<<x<<" "<<y<<endl;
  return 0;
}
```

Si ejecutamos el siguiente programa mostrará en la pantalla lo siguiente:

```
5 23
35 35
35 23
```

Ejercicio

Copia el ejemplo que se incluye en el material y pruébalo.

Sigue el programa que se muestra a continuación e indica qué valores se mostrarán en la pantalla

```
#include <iostream.h>
int main()
{
   int x, y, *p, *q;
   p = &x;
   *p = 12;
   q = &y;
   *q = 23;
   cout << *p <<" "<<*q<<endl;
   *p = *q;
   cout << *p <<" "<<*q<<endl;
   q = NULL;
   cout<<*p<<endl;
   cout<<x<<" "<<y<<endl;
   return 0;
}
```

ver solucion

Ligas sugeridas

http://www.cplusplus.com/doc/tutorial/

http://www.cs.wustl.edu/~schmidt/C++/

Regresar

Siguiente módulo