

Punteros y referencias

Breve introduccion por Mauricio Benitez mauriciobenitezp@fiuna.edu.py

Una variable de referencia es una “referencia” a una variable existente, y se crea con el operador &

```
int numero = 5; //variable numero
int &edad = numero; //referencia a la variable numero
```

Ahora podemos usar tanto la variable **numero** como la referencia **edad** para referirnos el valor de la variable **numero**.

```
int numero = 5;
int &edad = numero;
cout << numero << endl; //Imprime 5
cout << edad << endl; //Imprime 5 tambien
```

Para el ejemplo anterior, el operador & se usa para crear una variable de referencia, pero tambien se puede usar para saber la direccion de memoria de una variable, que es donde esta guardada la variable.

Cuando creamos una variable en C++, estamos asignandole un espacio de memoria de nuestra computadora a la misma. Y cuando le damos un valor a la variable, guardamos ese valor en esa direccion de memoria.

Para acceder a el, basta con usar & y el resultado representa la direccion de memoria donde esta guardado el valor de la variable.

```
int numero = 5;
cout << &numero; //Imprime 0xf1925ffcb4 (Podria imprimir otro valor para ustedes)
```

Punteros

Anteriormente aprendimos que podemos obtener la **direccion de memoria** de una variable usando el operador &.

Un **puntero** es una variable que guarda como valor una **direccion de memoria**

Una **variable puntero** apunta a un valor del mismo tipo (int, float, etc) y se crea con el operador *.

La direccion de la variable con la que estamos trabajando se le asigna al puntero

```
int numero = 5; //Una variable int que guarda el 5
int* pt = &numero; //Variable puntero con el nombre pt que guarda la direccion de numero

//Imprime 5
cout << numero << endl;
//Imprime la direccion de memoria de numero 0xf1925ffcb4
cout << &numero << endl;
//Imprime la direccion de memoria de numero con el puntero 0xf1925ffcb4
cout << pt << endl;
```

IMPORTANTE El tipo de dato del puntero debe ser el mismo que el de la variable que estamos trabajando.

Usamos el operador & para guardar la direccion de memoria de la variable numero en el puntero pt

Obtener direccion de memoria y valor

Para el ejemplo anterior, se uso el operador * y el operador & para crear una variable puntero. La misma almacena como valor la direccion de memoria de otra variable. Sin embargo, tambien podemos usar el operador * para obtener el valor de la variable. (Es decir, obtener que es lo que tiene guardado esta direccion de memoria)

```
int numero = 5;
int* pt = &numero;
//Imprime la direccion de memoria de numero
cout << pt << endl;
//Imprime el valor de numero, que es 5
cout << *pt << endl
```

Como seguramente sean confusos los usos de * , cito los usos:

- Cuando se usa en la declaracion de una variable (int* numero) crea una **variable puntero**.
- Cuando no se usa la declaracion de una variable, sirve para devolver el valor guardado en la direccion de memoria correspondiente.

Cambiar el valor de un puntero

Se puede modificar el valor de un puntero, pero modificarlo tambien cambiaria la variable original.

```
int numero = 5;
int* pt = &numero;
*pt = 10; //Modifica el valor del puntero, que modifica la variable original
//Imprime 10
cout << *pt << endl;
//Imprime 10 tambien
cout << numero << endl;
```