Punteros (C++)

Agosto 2017

Tipos punteros

- Un tipo puede ser un tipo puntero o un tipo valor
- Los tipos punteros permiten el paso por referencia
- Son fundamentales para la creación estructura de datos dinámicas (listas, colas, pilas árboles)

Declaración

- Una declaración tipo puntero toma una de las siguientes formas
  - int *ptrid, id;
  - id = 8;
  - ptrid = &id; // puntero a ID
- Se puede declarar más de un puntero en la misma sentencia. En el ejemplo que sigue se ve la declaración de dos punteros a int.
- int *ptrY, *ptrX;
- ptrid es un puntero a int, mientras que la variable id es solo una variable del tipo int. Todo puntero debe ser precedido por un asterisco (*) en la declaración
- Para obtener o modificar el valor de la variable a la que apuntan se utiliza el operador de indirección (*)
- Los punteros, al ser variables deben ser declaradas como punteros antes de ser utilizadas

Operadores

- Existen dos operadores a tener en cuenta cuando trabajamos con punteros
- El operador de dirección (&) que devuelve la dirección de memoria de su operando y el operador de indirección (*) que devuelve un alias o valor al cual apunta el operando del puntero
- En el siguiente ejemplo vemos como se inicializa una variable X con el valor 15
- Luego se crea un puntero a int y por último el puntero pasa a apuntar a la variable X. Esto es, ptrX es un puntero a X

  - int X = 15;
  - int *ptrX;
  - ptrX = &X;

Ejemplo

Para acceder a la dirección de memoria se utiliza &id, y sería igual a 0xA002

id = 8

Para acceder al valor indirecto o referenciado se utiliza el asterisco:
(*ptrid) lo que regresaría el valor de 8

ptrid = 0xA002

Dirección de memoria	0xA001	0xA002	0xA003	0xA004	0xA005	0xA006	0xA007	0xA008
Variables		id				ptrid		
Valores		8				0xA002		

```
int *ptrid, id;
id = 8;
ptrid = &id; // puntero a id
cout << "El valor de ptrid es " << *ptrid << endl; // mostraría 8
cout << "La dirección de memoria referenciada por ptrid es " << ptrid << endl; // mostraría 0xA002
cout << "La dirección de memoria de ptrid es " << &ptrid << endl; // mostraría 0xA006
```

Punteros en los vectores

- Los vectores son punteros constantes
- Un vector sin subíndice es un puntero al primer elemento del vector
- Una matriz es un vector de vectores (Ej: int M[3][3];) de manera que en cada elemento del primer vector "se asocia a" otro vector, pudiendo hacer así referencia a filas y columnas

  - int X[15];
  - int *ptrX;
  - ptrX = X; // ptrX recibe la dirección del primer elemento (0) de X

- Así como también podría escribirse
  - int X[15];
  - int *ptrX;
  - ptrX = &X[0]; // ptrX es igual a la dirección del primer elemento de X

- Se pueden utilizar distintos elementos del vector teniendo en cuenta la sintaxis de punteros
  - int X[15], Y, *ptrX;
  - ptrX = X;
  - Y = *(ptrX + 7);