Lenguaje de Programación

C++

Apuntadores y Alocación dinámica

José Luis Alonzo Velázquez

Universidad de Guanajuato

Sesión 38

Variables en memoria

En C++ guardamos la misma organización de las variables en memoria:

- Variables locales en M. PILA

Variables en memoria

En C++ guardamos la misma organización de las variables en memoria:

- Variables locales en M. PILA
- Variables globales en M. ESTÁTICA

Variables en memoria

En C++ guardamos la misma organización de las variables en memoria:

- Variables locales en M. PILA
- Variables globales en M. ESTÁTICA
- Variables estáticas en M. ESTÁTICA

Variables en memoria

En C++ guardamos la misma organización de las variables en memoria:

- Variables locales en M. PILA
- Variables globales en M. ESTÁTICA
- Variables estáticas en M. ESTÁTICA
- Datos alocados dinámicamente en M. MONTÍCULO

Variables Globales

Se pueden acceder desde otros archivos compilados en la misma aplicación, con la palabra llave **extern**: En file1.cpp: int myGlobalVariable; En file2.cpp : extern int myGlobalVariable;

Variables Globales

Cuando hay conflicto entre nombres de variables locales y de variables globales, a priori es la variable local que es considerada. Para utilizar explícitamente la variable global, se puede usar la notación ::var (también es valido para funciones/métodos, dentro de clases):

Ejemplo

```
int x=5;
int func(int n){
    int x ;
    x = 1 ;
    ::x = n+1;
}
```

Alocación dinámica

La alocación dinámica permite reservar pedazos de memoria en una zona especifica de la memoria (el montículo, o heap), al momento de la ejecución, según las necesidades.
Espacio reservado a través de un apuntador inicializado por funciones como malloc.
Espacio no liberado automáticamente (como lo de la pila) : uso de funciones explícitas de liberación, como free.

Apuntadores(punteros)

Un puntero es una variable que contiene una dirección de memoria. Normalmente, esa dirección es la posición de otra variable de memoria. Si una variable contiene la dirección de otra variable, entonces se dice que la primera variable apunta a la segunda.

Apuntadores(punteros)

Un puntero es una variable que contiene una dirección de memoria. Normalmente, esa dirección es la posición de otra variable de memoria. Si una variable contiene la dirección de otra variable, entonces se dice que la primera variable apunta a la segunda.

Sintaxis

TIPO * nombre_puntero ;

Apuntadores(punteros)

Un puntero es una variable que contiene una dirección de memoria. Normalmente, esa dirección es la posición de otra variable de memoria. Si una variable contiene la dirección de otra variable, entonces se dice que la primera variable apunta a la segunda.

Sintaxis

TIPO * nombre_puntero ;

Ejemplo

```
char *pchar;
```

Operadores de memoria

Existen dos operadores especiales de punteros: & y *. El operador de dirección (&) devuelve la dirección de memoria de su operando. El operador de indirección (*) devuelve el contenido de la dirección apuntada por el operando.

Asignación de punteros

Como en el caso de cualquier otra variable, un puntero puede utilizarse a la derecha de una declaración de asignación para asignar su valor a otro puntero. Por ejemplo:int x;

```
int *p1,*p2;
p1=&x;
p2=p1;
```

Observación

Tanto p1 como p2 apuntan a x.

Alocación dinámica en C

Típicamente, para un arreglo de 100 enteros:

```
int isize = 100;
int *i_ptr = (int *)malloc(isize *sizeof(int));
. . .
free(i_ptr) ;
```

free necesario para evitar fugas de memoria.

Alocación dinámica en C++

Otro sistema para la alocación dinámica, tal vez mas intuitivo que usar, con las palabras llaves new y delete :

```
int isize = 100;
int * i_ptr = new int [isize];
. . .
delete[ ] i_ptr;
```

delete necesario para evitar fugas de memoria.

Asignación ()

Tenemos que tener cuidado con lo siguiente

```
{
int* ip = 0;
ip = new int;
int* jp = new int(13);
[...]
delete ip;
delete jp;
}
```

 Como Programar en C/C++, Deitel (Prentice Hall), 2da Edición.

 Programming Principles and Practice Using C++, Bjarne Stroustrup.

 http://www.codeblocks.org

 http://www.wxwidgets.org

 (O'Reilly) Practical C Programming (3rd Edition)

 http://www.cplusplus.com