INFORMÁTICA INDUSTRIAL

PROGRAMACIÓN BÁSICA C++ (III)

M. Abderrahim, A. Castro, J. C. Castillo
Departamento de Ingeniería de Sistemas y Automática

uc3m | Universidad Carlos III de Madrid

10. Memoria Dinámica

Memoria dinámica

```
int * funcion (….)
{
    int * resultado = new int[5]; // Creación dinámica
        //hacer algo;
    return resultado;
}
```

- C++ almacena las variables locales en una pila (STACK), pero el operador “new” reserva memoria de almacenamiento dinámico (HEAP)
- Cuando se crea un array de forma dinámica, su tamaño se hace en el tiempo de operación
- Cuando se crea un array, su tamaño debe de ser siempre conocido en tiempo de compilación (es decir, no es un valor variable)

Uso de las memorias

Heap

- No está limitada (salvo la memoria disponible del sistema)
- Gestión dinámica de la memoria
  - malloc/calloc/realloc/**new** => free/**delete**
  - OJO. **memory leak** => memoria no liberada no puede ser reasignada
- Menos rápida
- Accesible desde cualquier función en en cualquier parte del código
- Variables de gran tamaño, dinámicas o que se van a utilizar en varias funciones

Stack

- Almacena variables temporales o locales
- Funcionamiento FILO
- Gestión por el sistema
- Limitada en tamaño
  - OJO. **stack overflow** (desbordamiento del stack) => CRASH!
- Muy rápida
- Cuando una función termina, todas sus variables son eliminadas.

Video

Datos

- Variables globales
- Variables estáticas

Código

- Las intrucciones

Memoria dinámica

delete pValue; //Puntero
delete [] list; //Array

- Solo se puede utilizar el comando “delete” con el puntero que apunta a la memoria que ha sido creada mediante el comando “new”

```
int main()
{
	int * b = new int;
	char* c = new char[12];

	*b = 5;
	delete b;
	delete [] c;
}
```

Memoria dinámica

```
int *p = new int;  p→0013FF60
*p = 45;
p = new int;  p→0013FF64
```

- Este procedimiento implica un problema de pérdida de memoria
- La posición original de memoria en la que se almacena el valor 45 (es decir, 0013FF60) no es accesible (debido a que se ha asignado nuevo espacio de memoria al mismo puntero p en 0013FF64) y la memoria original no puede ser borrada.

INFORMÁTICA INDUSTRIAL

PROGRAMACIÓN BÁSICA C++ (III)

M. Abderrahim, A. Castro, J. C. Castillo
Departamento de Ingeniería de Sistemas y Automática

uc3m | Universidad Carlos III de Madrid