

1.4.- Manejo de memoria

1. Memoria Estática
2. Memoria Dinámica
3. Memoria Automática

Memoria Estática

Siempre asigna espacio de memoria antes de utilizar los datos concretos.

Existirá memoria en el tiempo de ejecución del programa que nunca será utilizada.

Es la memoria que se reserva en el momento de la compilación antes de comenzar a ejecutar el programa. Los objetos son creados al iniciar el programa y destruidos al finalizar el mismo. Mantienen la misma localización en memoria durante todo el transcurso del programa hasta que son destruidos.

Los objetos administrados de este modo son: variables globales, variables estáticas de funciones, miembros static de clases y literales de cualquier tipo.

El inconveniente de la reserva estática es que la cantidad de memoria se reserva siempre antes de conocer los datos concretos del problema.

Tampoco se adapta bien a la memoria real disponible del ordenador en que se está ejecutando el programa.

¿Que podemos encontrar relacionado con el manejo de memoria estática?

- Variables estáticas
- Variables globales
- Miembros estáticos de clases
- Literales de cualquier tipo

Es posible que no se pueda adaptar de manera correcta a la memoria real de la computadora en la que se ejecuta el programa.

No es posible destinar cantidad de bytes exactamente en almacenamiento de manera exacta a procedimientos y funciones recursivas.

Memoria Dinámica

Se realiza en tiempo de ejecución del programa después de haber leído los datos y conocer del tamaño exacto de memoria a solicitar. Tiene mejor adaptación el proceso de solución de necesidades sin desperdiciar.

Es aquella que se reserva en tiempo de ejecución después de leer los datos y de conocer el tamaño exacto del problema a resolver. El sitio donde se almacenan los objetos se le denomina HEAP = MONTÍCULO pero el sitio preciso donde se

encuentra tal montículo depende del compilador y el tipo de puntero utilizado en la reserva de memoria dinámica.

Puntero (apuntador): un puntero o apuntador es un tipo especial de variable que almacena el valor de una dirección de memoria la cual puede ser de una variable individual, de un elemento de un arreglo, una estructura u objeto de una clase y se anota de la siguiente manera:

Tipo de apuntador + nombre de la variable.

- `Int * Pint;` puntero a un entero.
- `Char * Pchar;` puntero de carácter.
- `Fecha * Pfecha;` puntero objeto de la clase fecha.