

TAREA #4 MANEJO DE MEMORIA DINÁMICA Y ESTÁTICA

CINTHIA GUADALUPE OLIVAS CALDERON NO.17212165

La memoria (también llamada almacenamiento) se refiere a parte de los componentes que forman parte de una computadora, Son dispositivos que retienen datos informáticos durante algún intervalo de tiempo. Las memorias de computadora proporcionan unas de las principales funciones de la computación moderna, la retención o almacenamiento de información.

Memoria estática

La memoria estática es la que se reserva al momento de compilación antes de comenzar a ejecutarse el programa. Los objetos son creados en ese momento y destruidos al final del programa. Mantiene la misma de localización en memoria durante todo el transcurso del programa. Los objetos administrados de este modo son: variables globales, variables estáticas de funciones, miembros static de clases y literales de cualquier tipo.

La gestión de la memoria estática es la más sencilla. De hecho, algunos lenguajes de programación antiguos, como las primeras versiones de FORTRAN, únicamente tienen reserva estática de memoria. Las principales ventajas son la sencillez de implementación y que los requerimientos de memoria del programa son conocidos una vez compilado el mismo.

Sin embargo, existen bastantes inconvenientes:

- El tamaño de los objetos debe ser conocido en tiempo de compilación: no se puede trabajar con objetos de longitud variable.
- Es difícil para el programador definir el tamaño de las estructuras que va a usar: si son demasiado grandes, desperdiciará memoria; si son pequeñas, no podrá utilizar el programa en todos los casos.
- Sólo puede haber una instancia de cada objeto: no se pueden implementar procedimientos recursivos. Por estas razones, casi todos los lenguajes de programación tienen además la posibilidad de gestionar parte de la memoria de manera dinámica.

Su principal característica es que ocupan solo una casilla de memoria, por lo tanto, una variable simple hace referencia a un único valor a la vez, dentro de este grupo de datos se encuentra: enteros, reales, caracteres, booleanos, enumerados y subrangos (los últimos no existen en algunos lenguajes de programación)

Memoria con reserva dinámica

La memoria dinámica se refiere a aquella memoria que no puede ser definida ya que no se conoce o no se tiene idea del número de la variable a considerarse, la solución a este problema es la memoria dinámica que permite solicitar memoria en tiempo de ejecución, por lo que cuanto más memoria se necesite, más se solicita al sistema operativo

. El sistema operativo maneja la memoria gracias al uso de punteros, por la misma naturaleza del proceso nos impide conocer el tamaño de la memoria necesaria en el momento de compilar.

La memoria que ocupan los objetos con tiempo de vida no predecible a partir del código fuente se gestiona dinámicamente: cuando se crea un objeto en tiempo de ejecución, se pide memoria para él, y cuando se elimina, la memoria se libera.

En algunos lenguajes encontramos características que necesitan una gestión de memoria con reserva dinámica:

- Los operadores new y dispose del PASCAL,
- la creación de elementos de listas en LISP y su liberación automática (garbage collection) por ausencia de referencias a la memoria,
- las llamadas a funciones de biblioteca malloc y free del C,
- la creación de objetos en Java y su eliminación cuando ninguna variable mantiene referencias a dichos objetos,
- los operadores new y delete o las llamadas implícitas que comportan la creación y destrucción de objetos en C++, etc.

La zona de memoria destinada a atender las demandas de memoria dinámica se denomina heap.

¿Para qué sirve?

- Permite reservar/liberar memoria bajo demanda – Libera al programador de saber dónde colocar sus estructuras en memoria.
- Evita límites por culpa de variables estáticas.
- Facilita implementar estructuras de datos dinámicas como listas, árboles, en el que no se conoce a priori su tamaño final.
- Mejor aprovechamiento de la memoria.

– Operaciones

- Reservar (malloc)
- Liberar (free)

Así como existen estructuras de datos estáticas (arreglos), también existen estructuras de datos dinámicas (listas y árboles), estas últimas son generadas a partir de un tipo de dato conocido como referencia (dirección de memoria). Para utilizar las referencias se requiere de un elemento llamado nodo.

Referencias:

- *Procesadores de lenguaje Organización y gestión de la memoria. Universitat Jaume* (2006-2007)
(<http://repositori.uji.es/xmlui/bitstream/handle/10234/5915/memoria.apun.pdf;jsessionid=899370302D579FA75D40D3EDFB24E730?sequence=1>)
- *Yolanda Becerra Fontal, Juan José Costa Prats. (2017). Gestión de Memoria. 2017, de Facultat informàtica de Barcelona* Sitio web:
<http://docencia.ac.upc.edu/FIB/grau/SO2/documents/tema6.pdf>