Manejo de la memoria estatica y dinamica

Memoria Estática:

La forma más fácil de almacenar el contenido de una variable en memoria en tiempo de ejecuciones en memoria estática o permanente a lo largo de toda la ejecución del programa.No todos los objetos (variables) pueden ser almacenados estáticamente.Para que un objeto pueda ser almacenado en memoria estática su tamaño (número de bytes necesarios para su almacenamiento) ha de ser conocido en tiempo de compilación, como consecuencia de esta condición no podrán almacenarse en memoria estática:* Los objetos correspondientes a procedimientos o funciones recursivas, ya que en tiempo de compilación no se sabe el número de variables que serán necesarias.* Las estructuras dinámicas de datos tales como listas, árboles, etc. ya que el número de elementos que las forman no es conocido hasta que el programa se ejecuta.Las técnicas de asignación de memoria estática son sencillas. A partir de una posición señalada por un puntero de referencia se aloja el objeto X, y se avanza el puntero tantos bytes como sean necesarios para almacenar el objeto X.La asignación de memoria puede hacerse en tiempo de compilación y los objetos están vigentes desde que comienza la ejecución del programa hasta que termina.En los lenguajes que permiten la existencia de subprogramas, y siempre que todos los objetos de estos subprogramas puedan almacenarse estáticamente se aloja en la memoria estática un registro de activación correspondiente a cada uno de los subprogramas.Estos registros de activación contendrán las variables locales, parámetros formales y valor devuelto por la función.

Memoria 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 cuanta 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.

Un dato importante es que como tal este tipo de datos se crean y se destruyen mientras se ejecuta el programa y por lo tanto la estructura de datos se va dimensionando de forma precisa a los requerimientos del programa, evitándose así perder datos o desperdiciar memoria si hubiéramos tratado de definir la cantidad de memoria a utilizar en el momento de compilar el programa.

Cuando se crea un programa en el que es necesario manejar memoria dinámica el sistema operativo divide el programa en cuatro partes que son: texto, datos (estáticos), pila y una zona libre o heap. En la última parte es donde queda la memoria libre para poder utilizarla de forma dinámica. En el momento de la ejecución habrá tanto partes libres como partes asignadas al proceso por lo cual si no se liberan las partes utilizadas de la memoria y que han quedado inservibles es posible que se «agote» esta parte y por lo tanto la fuente de la memoria dinámica. También la pila cambia su tamaño dinámicamente, pero esto no depende del programador sino del sistema operativo.

Referencias:

https://es.scribd.com/doc/36911073/Memoria-Estatica

https://es.wikipedia.org/wiki/Memoria_din%C3%A1mica_(programaci%C3%B3n)