

Uso de la memoria estática

Para cuando un programa comienza a ejecutarse, debe haber algunos bloques específicos de memoria reservados para su uso que no pueden ser traspasados por ningún otro programa, ni por el sistema, ni siquiera por el programa mismo. Esto incluye, por ejemplo, la memoria que contiene el código del programa. Si bien es posible (en lenguaje de máquina) escribir un programa que pueda modificar su propio código, esta es una práctica muy peligrosa y nunca debería emplearse. No es posible hacer esto en la mayoría de los lenguajes de alto nivel como Modula-2.

Además, todas las variables nombradas en la sección de declaración deben tener memoria específica reservada para sus contenidos, y esta acción no puede ser controlada o modificada de ninguna manera por el programador, excepto al declarar más o menos variables en primer lugar. La memoria en cuestión no puede ser reubicada en otro lugar o expandida o contraída.

Las variables estáticas son aquellas que se crean en la sección de declaración de un programa y continúan existiendo (ya sean visibles o no) y requieren ese espacio hasta su conclusión. Su espacio se asigna al comienzo de la ejecución del programa.

Memoria dinámica y el montón - Dinámica controlada por programa

La descripción hasta ahora es solo una explicación de lo que ya se ha hecho; no ofrece nada nuevo. En particular, todos los tipos variables empleados hasta ahora deben tener una asignación de memoria predeterminada o estática. Si el programador decide que la cantidad máxima de estudiantes en una clase para un programa de calificaciones será 100, el programa simplemente no permitirá la entrada al número 101.

Sin embargo, la vida real es diferente. Tome reservas aéreas, por ejemplo. El número de registros de pasajeros activos puede crecer y reducirse de acuerdo con la hora del día, el día de la semana, la temporada del año y si el asistente viste de verde o sopla viento del Este. El punto es que no siempre es posible predeterminar el número máximo de elementos de un tipo de datos que representa una reserva con anticipación.

Por lo tanto, es natural preguntar si es posible crear algunos tipos de datos que no tienen un número fijo de instancias, ni una asignación de memoria predeterminada, sino que pueden crearse instancias y la memoria asignada por el programa en ejecución, de modo que el máximo número de entidades está limitado solo por la memoria disponible en lugar de por el programa.

Lo es, y el propósito principal de este capítulo es mostrar cómo se hace esto en Modula-2. Primero, la discusión precedente se formaliza en la siguiente definición:

Las variables dinámicas son aquellas que pueden tener el espacio asignado a ellas según sea necesario en algún momento durante la ejecución de un programa o procedimiento y que también pueden ser eliminadas y el programa puede devolver su espacio al sistema.

El único lugar donde dicha actividad puede tener lugar en el modelo de memoria que se discute aquí está en el tramo de memoria desde la parte superior de la pila hasta el final de la región asignada al programa. Dentro de esta región, algunos lenguajes (incluido Modula-2) permiten la asignación manual y la desasignación de memoria bajo el control del programa.

La región de memoria sobre la pila y en la que puede tener lugar la asignación dinámica controlada y la desasignación de memoria se denomina montón .

Para emplear el montón, un programa debe tener:

- una o más variables de puntero estático para mantener las ubicaciones de las variables dinámicas
- un método para obtener memoria de pila y devolverla (asignar / desasignar)
- un método para asignar la dirección de dicha memoria a la variable del puntero estático
- un método para referirse a la memoria dinámica a través de la variable estática

Los medios por los cuales se lograron los puntos primero y último ya han sido cubiertos. Las variables de puntero Modula-2 se utilizan como variables estáticas para contener las direcciones de los trozos dinámicos de memoria, y se eliminan para referirse a los datos reales. Los detalles de asignación de memoria y desasignación en Modula-2 son el tema de la siguiente sección. Sin embargo, en este punto, el mapa de memoria en este modelo de gestión de memoria podría parecerse a la figura 12.4, en la que se supone que el programa tiene dos variables de puntero estáticas para contener ubicaciones de memoria dinámica, y ya ha obtenido la memoria y asignado el direcciones a esas variables.

Bibliografía

<http://www.csc.twu.ca/rsbook/Ch12/Ch12.4.html>