

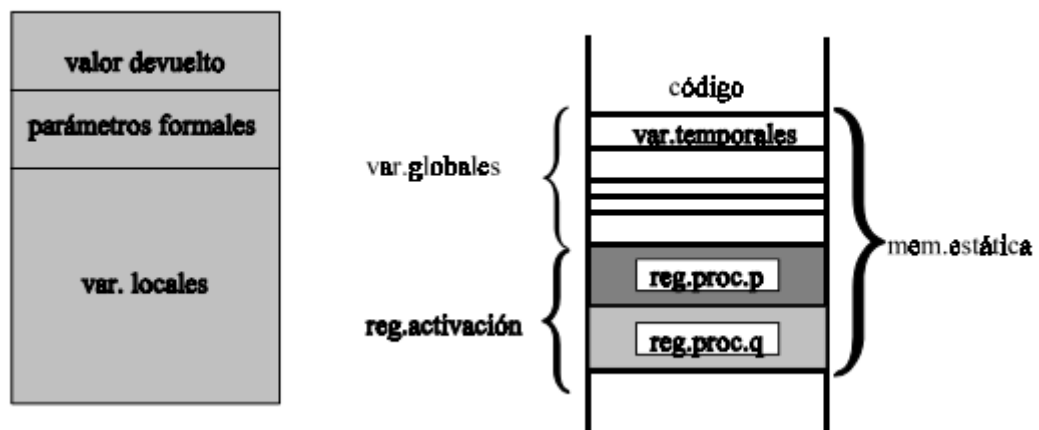
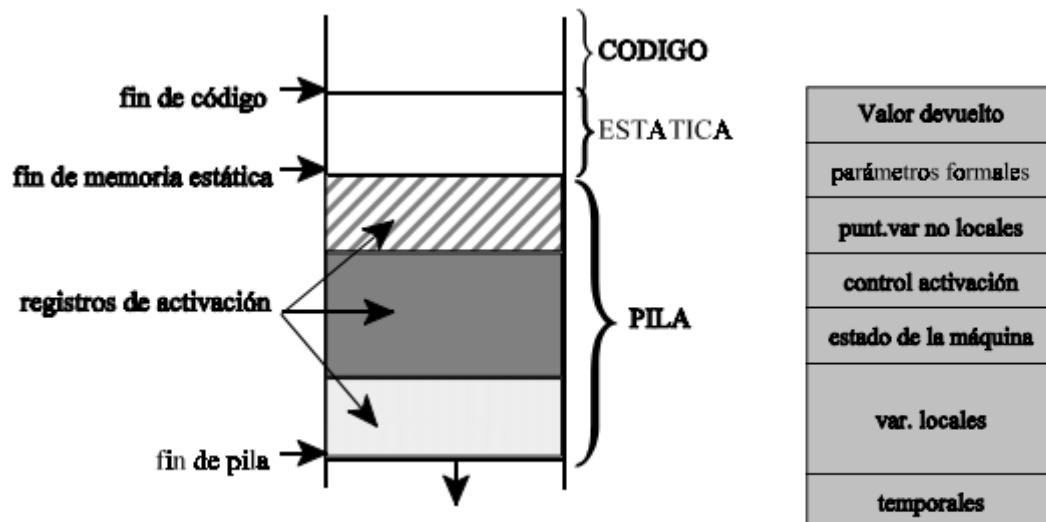
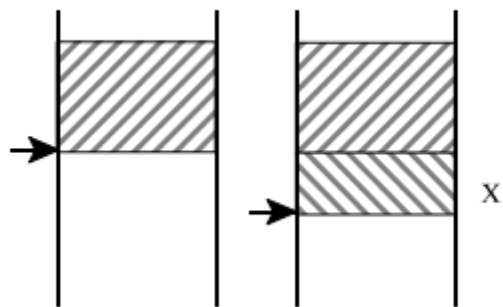
Manejo de memoria estática

La forma más sencilla de almacenar el contenido de una variable en memoria, en tiempo de ejecución, es hacerlo en la memoria estática. Así, el almacenamiento de dichas variables será permanente (durante la ejecución del programa). Por ello, resulta obvio que los datos etiquetados como constantes y las variables globales de un programa tengan asignada la memoria necesaria durante toda la ejecución del programa. Sin embargo, no todas las variables pueden almacenarse estáticamente. Para que una variable pueda ser almacenada en memoria estática, es necesario conocer su tamaño (número de bytes necesarios para su almacenamiento) en tiempo de compilación.

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.
- Solo 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. La gestión de esta memoria se puede hacer íntegramente en tiempo de compilación. Para acceder a un objeto, el programa únicamente necesita saber en qué dirección se encuentra. Estas direcciones se pueden asignar secuencialmente. Basta con que el compilador tenga anotada la primera dirección libre del bloque de memoria estática y la vaya actualizando sumando la talla de los objetos que va reservando.



Referencias

1. Ingeniería Informática: Procesadores de lenguaje, universitat Jaume, Pag. 2.
2. Estructura de datos: Algoritmos, abstracción y objetos, Luis Joyanes, McGraw-Hill, 1998, España, Pag. 126.
3. Heileman, Gregory L, Estructura de datos, algoritmos y programación orientada a objetos, Madrid, McGraw-Hill, 1997.
4. Franch Gutierrez, Estructura de datos, especificación, diseño e implementación, Barcelona, Ediciones UPC, 1994.
5. Collins, William J, Data Structures, An object-Oriented Approach, Addison-Wesley, 1992.