

#### 4.4 Administrador de memoria

La administración de la memoria es un proceso hoy en día muy importante, de tal modo que su mal o buen uso tiene una acción directa sobre el desempeño de memoria. En general un ensamblador tiene un administrador de memoria más limitado que un compilador; en la mayoría de los lenguajes de programación el uso de punteros no estaba vigilado por lo que se tienen muchos problemas con el uso de memoria. Los lenguajes más recientes controlan el uso de punteros y tienen un programa denominado recolector de basura que se encarga de limpiar la memoria no utilizada mejorando el desempeño.

La memoria principal puede ser considerada como un arreglo lineal de localidades de almacenamiento de un byte de tamaño. Cada localidad de almacenamiento tiene asignada una dirección que la identifica

Se distinguen los siguientes propósitos del sistema de administración de memoria:

- *Protección.*

Si varios programas comparten la memoria principal, se debería asegurar que el programa no sea capaz de cambiar las ubicaciones no pertenecientes a él. Aunque una acción de escritura puede tener efectos más graves que una de lectura, esta última tampoco debería estar permitida, para proporcionar algo de privacidad al programa.

- *Compartimiento.*

Este objetivo parece contradecir al anterior, sin embargo, a veces es necesario para los usuarios poder compartir y actualizar información (por ejemplo, en una base de datos) y, si se organiza la tarea de entrada a la misma, se puede evitar el tener varias copias de la rutina.

- *Reubicación.*

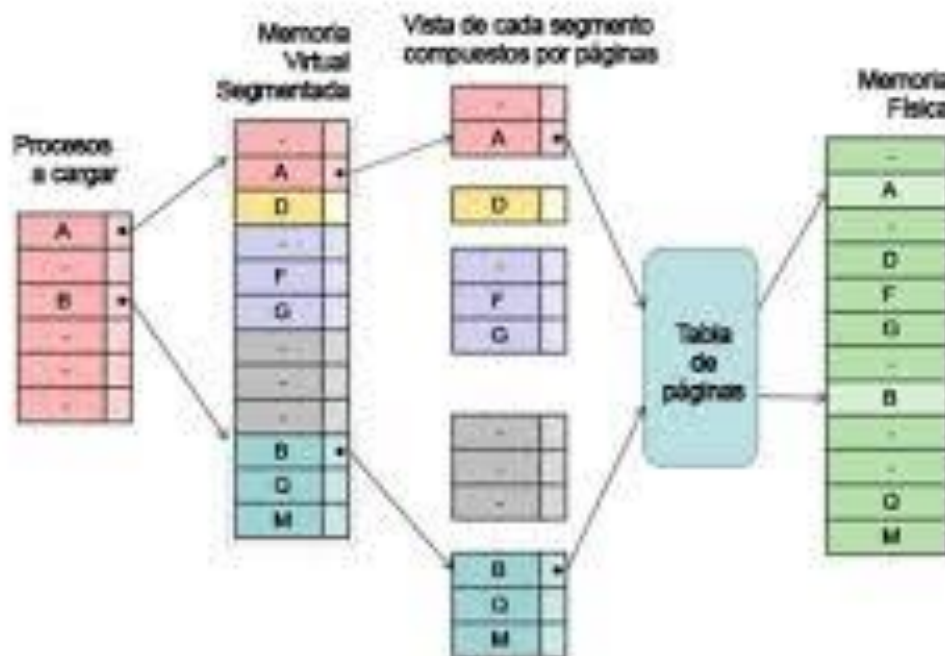
La técnica de multiprogramación requiere que varios programas ocupen la memoria al mismo tiempo. Sin embargo, no se sabe con anticipación donde será cargado cada programa por lo que no es práctico usar direccionamiento absoluto de memoria.

- *Organización física.*

Debido al costo de una memoria principal rápida, éste se usa en conjunto con una memoria secundaria mucho más lenta (y por consiguiente, barata) a fines de extender su capacidad.

- *Organización lógica.*

Aunque la mayor parte de las memorias son organizadas linealmente con un direccionamiento secuencial, esto difícilmente concuerde con el camino seguido por el programa, debido al uso de procedimientos, funciones, subrutinas, arreglos, etc



### Equipo Amarillo

La información investigada aquí es por parte de todo el equipo sin menospreciar el trabajo de alguno al igual que en el desarrollo de la página

## BIBLIOGRAFÍA

Lenguaje y Autómatas II. Unidad VI Generación de Código Intermedio. M.C. Juan Carlos Olivares Rojas. pág. 1-36. Año 2010.

Generación de código procesador de lenguaje. Lenguaje y autómata II. Universidad técnica de México pág. 1-28.

**Equipo Amarillo**

La información investigada aquí es por parte de todo el equipo sin menospreciar el trabajo de alguno al igual que en el desarrollo de la página