

# Gestión y administración de la memoria

La memoria es una **matriz de palabras o bytes direccionables** accesibles mediante una dirección única por la CPU, para la carga de instrucciones o datos, o su almacenamiento. Las direcciones que entiende el controlador de la memoria son direcciones absolutas. La conversión entre las distintas representaciones de las direcciones absolutas se denomina **vinculación**, y puede llevarse a cabo en cualquiera de las fases del programa.

En el **direccionamiento** se usan varios tipos de direcciones:

- **Dirección lógica:** Es la generada por la CPU (punto de vista del programa).
- **Dirección física:** Dirección real percibida por la unidad de memoria.

Si la vinculación se realiza en tiempo de compilación o de carga, entonces la dirección física = dirección lógica.

Si la vinculación se realiza en tiempo de ejecución, las direcciones lógicas y físicas difieren, y llamamos a las direcciones lógicas virtuales.

La necesidad de la gestión de la memoria se debe a que tenemos que:

- Ubicar los procesos en memoria.
- Subdividir la memoria para hacer sitio a varios procesos.
- Repartir eficientemente la memoria para introducir tantos procesos como sea posible.

La gestión debe **reubicar, proteger, compartir, organizar lógica y físicamente**.

En la reubicación, programador no conoce qué otros programas residirán en la memoria en el momento de la ejecución. Se deben traducir las referencias a la memoria encontradas en el código del programa a las direcciones físicas reales.

En la protección, el código de un proceso no puede hacer referencia a posiciones de memoria de otros procesos sin permiso. Esto debe comprobarse durante la ejecución; el sistema operativo no puede anticiparse a todas las referencias a la memoria que hará un programa. Solo el SO puede modificar los registros de base y límite.

En la compartición, se permite el acceso de varios procesos a la misma zona de la memoria principal.

Durante la organización lógica, la memoria está organizada como un espacio de direcciones unidimensional. La mayoría de los programas se organizan en módulos.

En la organización física, tenemos la **memoria principal** (rápida, costosa, pequeña, volátil...) y la **memoria secundaria** (más lenta, barata, gran capacidad, no volátil...). El flujo entre ambas es responsabilidad del S.O.

Estos han sido los requisitos de la gestión de memoria.

Existen técnicas para gestionar la memoria, estas se basan en la carga de programas en memoria principal para su ejecución en el procesador.

- **Particiones estáticas:** El S.O. ocupa una parte fija de la memoria, el resto está disponible para ser usada por los procesos.
  - **Particiones de igual tamaño:** Cualquier proceso con tamaño menor o igual al tamaño de la partición puede cargarse. Si todas las particiones están ocupadas el sistema operativo puede sacar un proceso de una partición (lo suspende).  
Los problemas son que puede que un programa no se ajuste a una partición. El programador de diseñar el programa mediante superposiciones. A parte, el uso de la memoria principal es ineficiente, cualquier programa pequeño ocupará una partición entera, esto se denomina **fragmentación interna**.

Sistema operativo 8 MB
8 MB
8 MB
8 MB
8 MB
8 MB

- **Particiones de distinto tamaño:** En comparación al modelo anterior, se reducen los problemas. Se pueden alojar programas más grandes sin superposición y se reduce la fragmentación interna. Los problemas son que caben menos programas grandes sin superposición y si hay muchos programas pequeños la fragmentación interna es muy grande en las particiones grandes. Existen 2 formas de gestionarlas, mediante una cola por partición, o mediante una única cola.

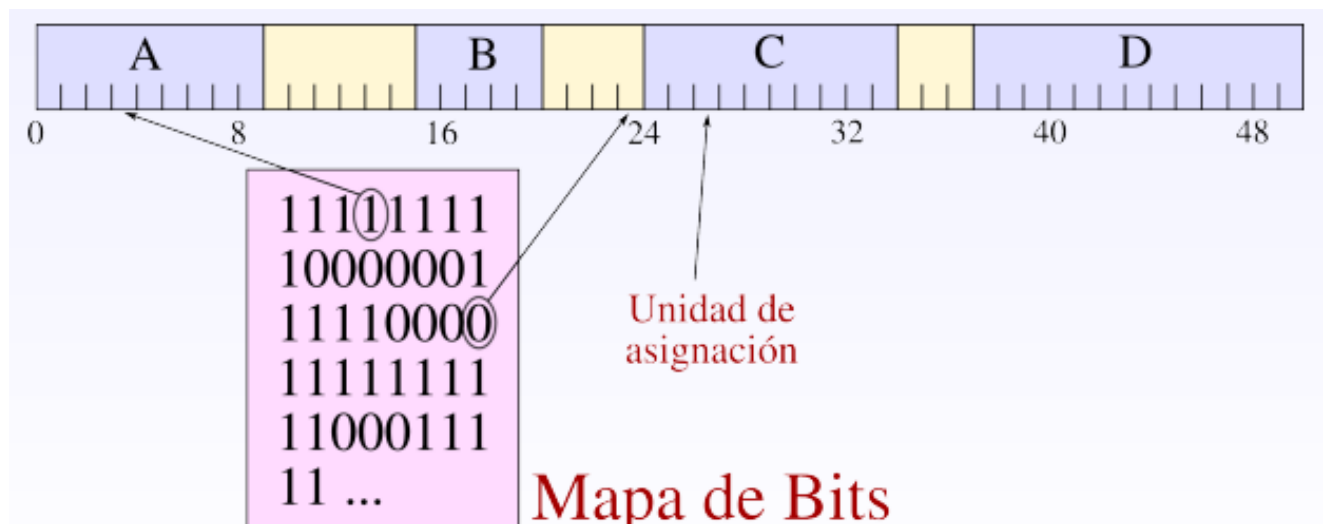
Si se usa una única cola, cuando se va a cargar un proceso se selecciona la partición más pequeña disponible. Como desventajas tenemos que el número y el tamaño de las particiones limitan el número de procesos y el tamaño máximo de procesos activos.

Si hay muchos procesos pequeños se produce la fragmentación interna. Si hay muchos procesos grandes, las particiones pueden resultar infrutilizadas.

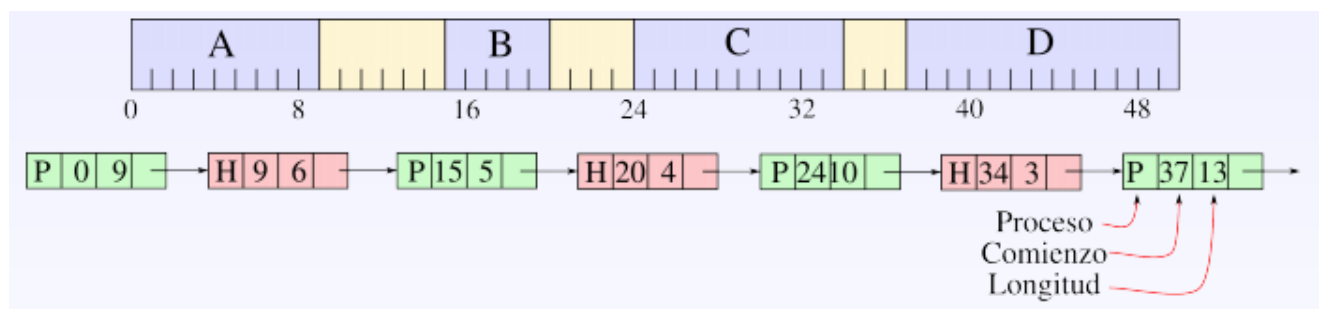
Si se usa una cola por partición, cada proceso se ubica en la partición de tamaño menor en la que quepa. Si la partición no está libre, espera en la cola correspondiente. Se minimiza la memoria desaprovechada dentro de cada partición, aunque pueden existir procesos suspendidos esperando en colas correspondientes a las particiones que mejor se ajustan a su tamaño habiendo otras particiones vacías.

- **Particiones dinámicas:** En este sistema tras la reubicación de procesos, se van generando huecos en memoria entre las distintas particiones que, unidos, serían útiles para albergar procesos, pero, separados no sirven. Este fenómeno se denomina **fragmentación externa**. Se debe usar la **compactación** para desplazar los procesos de modo que queden situados en particiones contiguas.

En este esquema, la memoria principal está dividida en unidades de asignación (bloques). Para conocer si están libres u ocupados, se utiliza un **mapa de bits**.

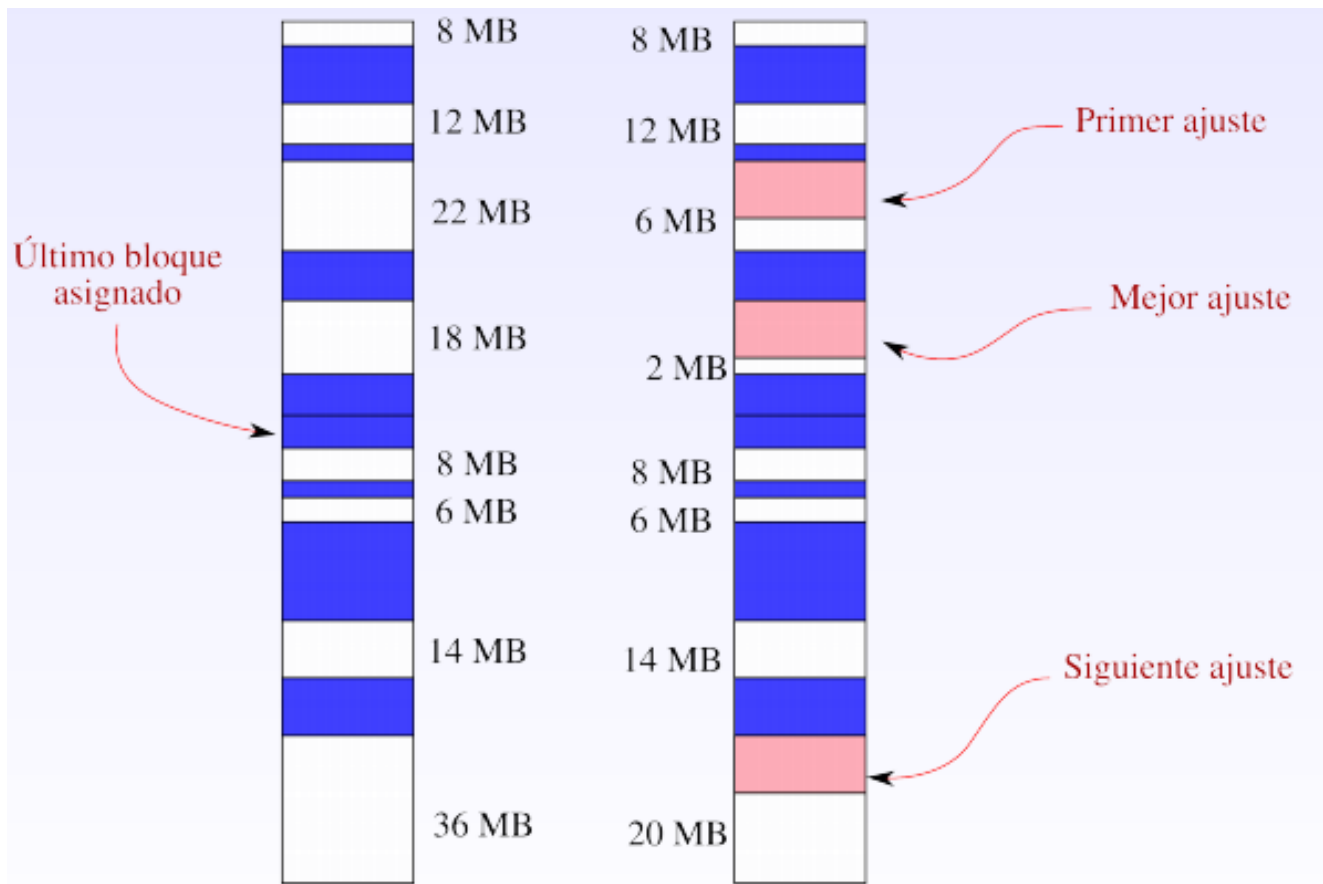


También se pueden usar **listas enlazadas**. Las listas tienen ventaja sobre el mapa ya que en la lista la búsqueda es más rápida, puesto que cada elemento de la lista contiene el tamaño del mismo. A parte, es fácil reorganizar la lista al introducir un nuevo proceso.

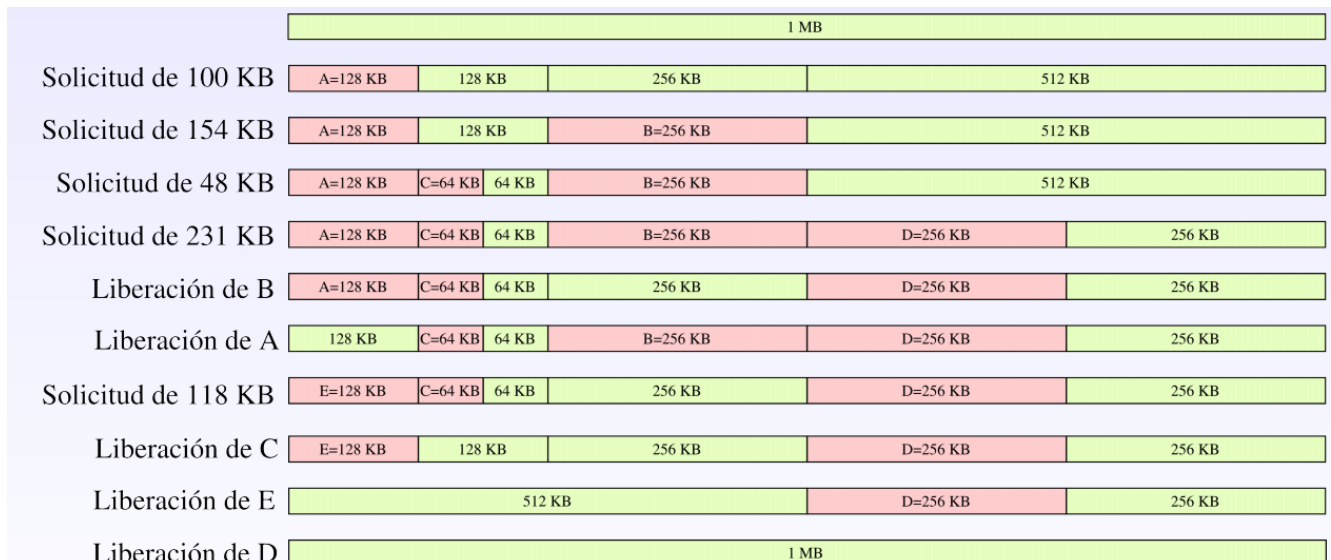


Otro tema es la **ubicación por ajuste**. Existen 3 métodos principales:

- **Best-fit**: Se elige el bloque de tamaño más próximo al solicitado. En general proporciona malos resultados.
- **First-fit**: El más rápido, en cuanto encuentra un hueco, lo usa.
- **Next-fit**: Busca a partir del último bloque asignado.

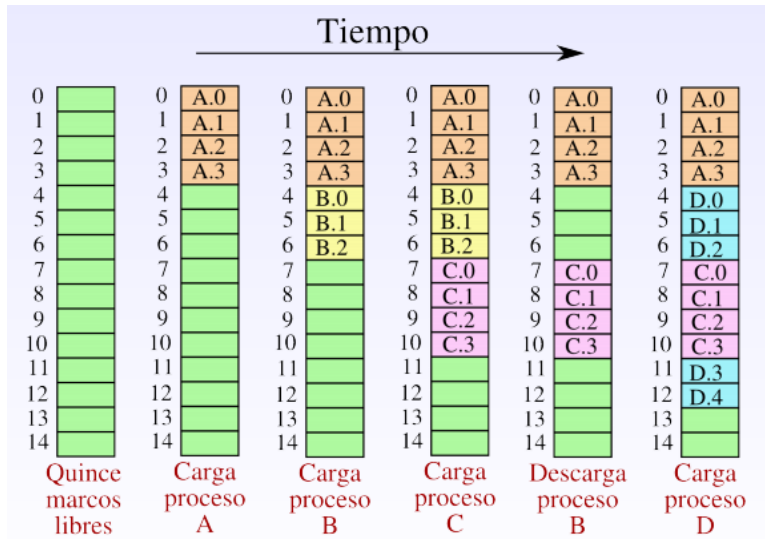


Otro método es el **sistema de colegas**.



(Errata: Color del hueco que deja B al liberarse: verde desde que se libera)

- **Paginación:** La memoria principal se encuentra dividida en pequeñas partes iguales de tamaño fijo. Cada una de esas partes se denomina **marco**. La imagen de cada proceso se divide en partes pequeñas con el mismo tamaño que el marco, a estas se las denomina **páginas**. El sistema operativo mantiene una tabla de páginas para cada proceso.



- **Segmentación:** La imagen de un proceso está formada por múltiples segmentos. No es necesario que todos los segmentos de todos los programas tengan la misma longitud, aunque existe una longitud máxima de segmento. Cada segmento se almacena entero de forma contigua en memoria principal y se crea una tabla de segmentos para cada proceso. Al emplear segmentos de distintos tamaños, la segmentación resulta similar a la partición dinámica.