

Gestión de memoria:

La memoria principal:

La memoria principal es un espacio de direcciones (un conjunto de celdas) a las que se puede acceder mediante su dirección: 0x0000 – 0xFFFF

Condiciones iniciales:

Para que un programa pueda ejecutarse debe encontrarse en la memoria principal.

En sistemas multiprogramados (multitarea) varios programas (o al menos parte de ellos) deben estar cargados en la memoria principal.

El sistema operativo para poder realizar sus funciones también debe estar cargado (al menos en parte) en la memoria principal.

Objetivo principal:

Permitir que varios procesos puedan ejecutarse de forma concurrente y segura.

Un proceso debe estar aislado del resto: no se le dejará acceder a las zonas de memoria en las que se ubica otro proceso (ni el S.O.).

El sistema operativo debe estar protegido de la acción de los procesos activos.

Objetivos secundarios:

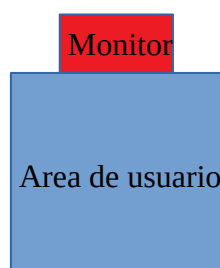
Debe aprovecharse de forma óptima. Minimizar la memoria desaprovechada.

No se debe penalizar el tiempo de ejecución de los programas. La gestión de memoria debe intentar realizarse de forma imperceptible.

Técnicas de gestión de la memoria principal:

Sistemas primitivos (proceso por lotes):

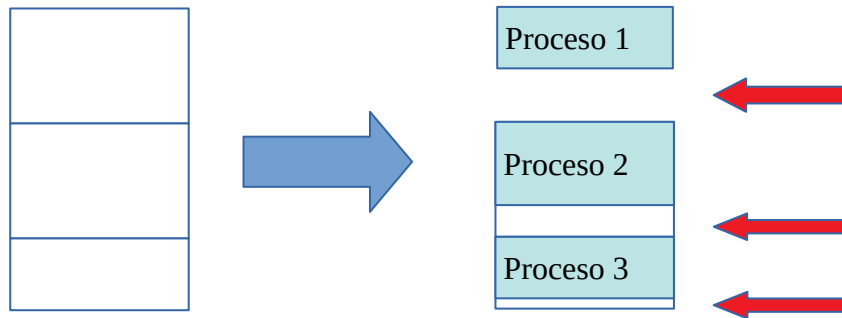
En sistemas monotarea, con el primitivo Monitor – Residente, se partía el espacio disponible en la memoria en dos partes: una para el sistema operativo (monitor) y otra para los programas de usuario.



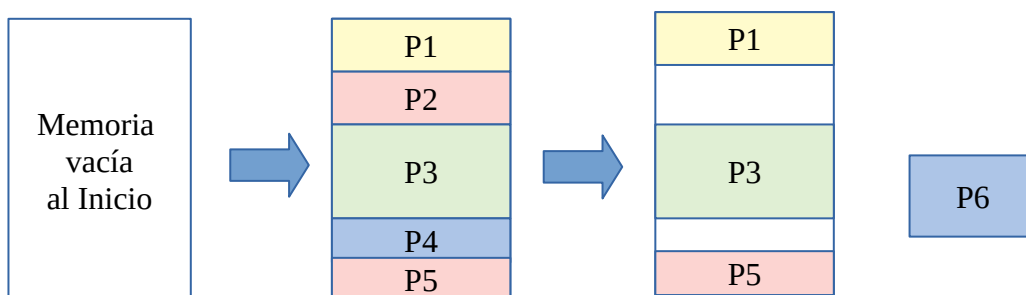
Sistemas multiprogramados:

Particiones de tamaño fijo: Se hacen varias particiones. No tienen por qué ser todas del mismo tamaño. No varían con el tiempo.

Fragmentación Interna: Se desaprovecha espacio dentro de cada partición (indicadas por las flechas en rojo). Los procesos suelen tener un tamaño inferior al de la partición.



Particiones de tamaño variable: Se hacen particiones a medida de los procesos. No se desaprovecha espacio inicialmente. Al finalizar unos procesos se libera el espacio utilizado y al arrancar unos nuevos con tamaños diferentes dan lugar a la aparición de pequeños espacios entre particiones que se acaban desaprovechando.



Si se quiere cargar un proceso P6 cuyo tamaño es mayor que P2 pero menor que $P2 + P4$ (el espacio que ha quedado libre) no se permitirá al no haber espacio CONTIGUO suficiente.

Mover los procesos de una zona a otra no es práctico. El sistema ocuparía su tiempo en monitorizar y reubicar procesos en lugar de ejecutarlos.

Segmentación:

Los procesos se dividen en segmentos que pueden ubicarse en zonas diferentes y no contiguas. Esto permitirá aprovechar mejor los huecos pequeños.

Paginación:

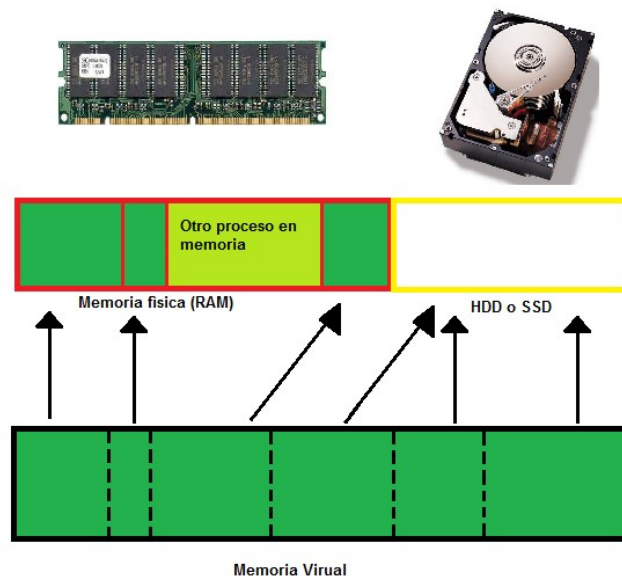
El espacio disponible en la memoria se divide en particiones pequeñas de tamaño fijo: las páginas. Cada página puede alojar fragmentos de un proceso. Si un proceso se almacena en varias páginas solo habrá fragmentación en la última de ellas:



Memoria Virtual con paginación a demanda

Los sistemas operativos actuales pueden compensar la falta de espacio en la memoria principal utilizando las unidades de almacenamiento como si se trataran de una parte de la memoria principal.

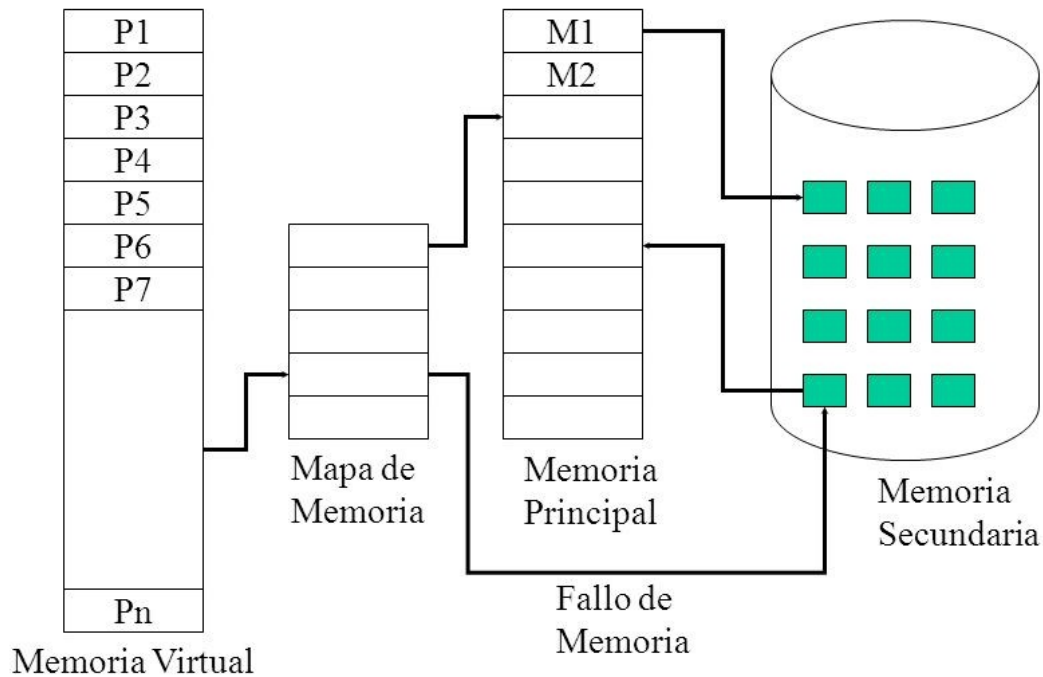
Estos sistemas crean una memoria virtual: un espacio de direcciones cuya capacidad es mayor que la capacidad de la memoria RAM. Esa memoria virtual se divide en páginas, que se almacenan físicamente en la RAM y aquellas que no caben ahí se guardan en una unidad de almacenamiento.



Esto plantea un problema: si un proceso intenta acceder a una instrucción o un dato que está en una página de las que se encuentran en el disco, ¿Qué ocurre?. Pues se da una situación en la que se necesita pasar esa página a la memoria principal pero previamente hay que hacerle hueco pasando una de las que allí se encuentran al disco. A esta operación se la llama **intercambio (o swapping)**.

La situación que da lugar a una operación de intercambio es conocida como **fallo de página**.

Gestión de Memoria Virtual



Algoritmos de reemplazo de páginas

A la hora de elegir cuál de las páginas que hay en RAM va a ser trasladada al disco tras un fallo de página se pueden seguir diferentes criterios:

FIFO: Se pasa al disco la más antigua, la que primero se cargó en la RAM

LRU: Se pasa al disco la que menos se ha usado recientemente ya que se supone que es la que menos se usará en el futuro.

Existen modificaciones basadas en las dos anteriores que mejoran su rendimiento.

Ver: https://es.wikipedia.org/wiki/Algoritmo_de_reemplazo_de_p%C3%A1ginas#%C3%93ptimo