**Mermoria ISO**

Administrador de memoria:

Como sun ombre indica es el encargado de administrarla

Posee un registro de las partes de memoria en uso

Asigna espacio en memoria para los procesos que lo necesitan.

Busca cargar en memoria la mayor cantida de procesos.

Direccion virtual:

Es la dirección que hace referencia dentro del proceso

Direccion física:

Hace referencia directa a la dirección de memoria RAM del proceso.

Se accede mediante el bus de direcciones.

Direccion lógica:

MMU: (Memory Managment Unit)  
 Se encarga de transformar las direcciones virtuales en física.

Se hace usando un registro de relocacion que guarda un entry point.

La direccon física es igual a la suma del entry point mas la dirección lógica

Division de memoria

Se busca que no haya problemas de ambigüedad de asignación de memoria.

Cada proceso tiene su partición.

Todo el proceso tiene que estar en memoria en particiones continuas.

Particiones fijas:

Primer Ajuste, Mejor Ajuste, Peor Ajuste, Proximo Ajuste

Genera fragmentación interna

Particiones dinámicas

Las particiones se van a ir creando o desapareciendo a demanda

Cada partición va a tener su tamaño especifico.

Genera fragmentación externa

Paginacion

Como solución a la fragmentación externa se crea la paginación

Se basa en dividir toda la memoria en **marcos** y a los procesos en **paginas.**

Siendo el tamaño de los marcos y paginas igual, generalmente 512 bytes.

El SO tiene una tabla de paginas para cada proceso, donde guarda la direccion d el marco que contiene cada pagina

**Todas las paginas deben estar en memoria , pero en lugares discontinuos.**

Las paginas se numeran desde el **0**, los marcos desde el **1**.

Para el acceso se usa la dirección dede la pagina + el desplazamiento (menor al tamaño de la pagina)

Si tenemos una dirección **lógica:** (Tm = tamaño pagina/marco)

El numero de pagina = dir lógica DIV Tm

El desplazamiento = dir Logica mod Tm

Direccion **FISICA** = dir Comienzo marco + desplazamiento

Si tenemos una dirección **física:**

El numero de marco = dir Fisica div Tm

` El desplazamiento = dir Fisica mod Tm

Direccion Logica = Tm + desplazamiento

Segmentacion:

Es una mejora de la paginacion.

Tiene una tabla de segmentos que tiene dirección base y una longitud limite

Cada dirección lógica guarda numero de segmento y desplazamiento

Si el desplazamiento es menor al limite se le suma la dirección base y se obtiene la dirección física.

Memoria virtual

**Page fault**: Ocurre cuando se quiere usar una pagina, la cual no esta cargada en memoria, por lo que se procede a cargarla

A la pagina que se le decide borrar de memoria se la llama pagina victima.

Si la pagina víctima fue modificada se deberá guardarla en el área de swap o dispositivo de paginación.

Algoritmos para elegir pagina victima:

Algoritmo FIFO:

La primera pagina que se varga es la primera en salir

Algortimo FIFO con segunda chance:

Se utiliza un bit de referencia (bit R, En hoja \*).

Cuando una pagina se agrega a la cola se pone con el bit en 0.

Cuando a una pagina ya en la cola se la vuelve a llamar, se le pone el bit en 1.

Cuando se requiera hacer una baja se busca la primer pagina con el bit en 0, y mientras busca va cambiendo las hojas con bit en 1 a bit en 0.

Tiene la desventaja que si una pagina fue referencia da muchísimas veces tiene la misma preferecia que una pagina que fue referenciada 1 vez.

Algortimo LRU:

Se puede ver como una mejora del FIFO.

Cada vez que vuelve a ser referenciada una pagina se pone a dicha pagina ultima en la lista.

Es decir se elije la que tiene la ultima referencia mas vieja

Paginacion Asignacion de marco:

Cuantos marcos se le asignara a cada proceso.

Asignacion fija:

Asignacion equitativo: A cada proceso se le asigna la misma cantidad de marcos.

Reparto proporcional: Se le asignan marcos en base a la cantidad de paginas de cada proceso

Asignacion Dinamica: El SO va cargando marcos de forma dinámica de acuerdo a los marcos que requiera

Alcanze de remplazo:

A la hora de elegir la victima se puede hacer

Remplazo global: Puede remplazar a cualquier proceso cargado en memoria.

Se usa con asignación dinámica, suele generar hyper-paginacion.

Remplazo local: La pagina victima se elige entre las paginas del mismo proceso. En este caso la hyper-paginacion solo afecta al propio proceso

Descargas asincrónicas:

Las paginas que son modificadas (en practica se marcan con m), no pueden ser borradas de una, por lo que se un o mas marcos como **marcos de descarga**.

La pagina que provoca el page fault se coloca en el marco de descarga, y el marco victima pasa a ser el nuevo marco de descarga.

De esta forma se da tiempo a que se guarde la pagina modificada

Performance:

Se busca la menor cantidad de page fault, ya que requiere ir al dispositivo de paginación que es lento.

Para eso existe una ecuacuion que define la probabilidad de page fault:

At = tiempo ded acceso a la pagina de tablas

Am tiempo de acceso a memoria real

Tf = tiempo de atención a un fallo de pagina

P = probabilida de page fault (entre 0 y 1)

TAE = Efective acces time

TAE = At + (1 – p) \* Am + P \* (Tf + Am)

Si la P es cercana a 1 es cuando existe la hyper-paginacion

Hyper-paginacion o trashing:

Ocurre cuando P llega a 1 o se acerca a 1.

Esto quiere decir que el SO se pasa mas tiempo cargando paginas o atendiento page fault que ejecutanto procesos.

Anomalia de Belady:

Se supone que si se asigna mas marcos a un procesos, no habría trashing. SAglvo en la anomalia de belady

Working set o modelo de localia:

Dice que las referencias a datos y programas dentro de un procesos tienden a agruparse, es decir que si se uso algo esprobable que se vuelva a usar.

Por eso se establece un conjunto de paginas (**working set**) del tamaño de un delta donde se guarda la referencia a las paginas mas recientes.

Si el delta es chivo no cubrira la localidad, tendrá muy pocas referencias.

Si el delta es muy grande va a tomar varias localidades, va a tener referencias viejas