Sistemas Operativos 2

Unidad 1: Administración de memoria

Memoria virtual

René Ornelis
julio de 2025

Contenido

Índice de figuras

Memoria virtual

1 Conceptos generales

Vimos en el capítulo anterior que el sistema operativo necesita administrar la memoria con un balance de eficiencia y usabilidad para el usuario. Esto es un objetivo bastante difícil de conseguir considerando que los procesos tienden a consumir cada vez más memoria y muchas veces piden memoria al sistema operativo que probablemente nunca utilizarán. Entonces si el sistema operativo sólo trabajará con la memoria que está disponible en RAM en muy poco tiempo o con muy pocos procesos llegará a quedarse sin memoria lo que no permitirá al usuario cargar más procesos, lo cual reduce la usabilidad del sistema ante los ojos del usuario.

Debido a esto se inventó la memoria virtual la cual básicamente consiste en el intercambio continuo de páginas de memoria entre la memoria principal y el disco duro, con el fin de permitir que el sistema muestre más memoria de la que realmente tiene y permita trabajar más procesos de los que cabrían en la memoria principal.

Esto también se tiene que hacer con precisión y cuidado del balance entre el rendimiento y la usabilidad, debido a que, si se abusa de este proceso, es decir hay muchos accesos a disco para bajar páginas o subir páginas de memoria, el impacto en el rendimiento del sistema será sensible porque el rendimiento del disco es mucho más lento que el acceso a memoria RAM.

Memoria virtual: Capacidad del sistema de permitir a los procesos utilizar más memoria que la disponible en RAM, a través de utilizar memoria secundaria (discos duros) para almacenar parte de la memoria en RAM, con énfasis en conservar un balance entre la usabilidad y rendimiento.

En esta unidad estudiaremos las diferentes técnicas y las desventajas de cada una de ellas para el manejo eficiente de la memoria virtual en los sistemas operativos.

Hiperpaginación: Exceso de fallos de página.

Fallo de página: Es el evento que un proceso accede una página que no está en memoria y se tiene que buscar en el disco duro (provocar un intercambio).

Desperdicio (trashing): Es cuando el procesador pasa más tiempo decidiendo qué páginas intercambiar y esperando respuesta del disco, que ejecutando procesos. Es una consecuencia directa de la hiperpaginación

Localidad temporal: la probabilidad condicional de que suceda de nuevo un evento dado que sucedió hace poco tiempo, es inversamente proporcional al tiempo transcurrido.

Localidad espacial: la probabilidad condicional de que suceda de nuevo un evento dado que sucedió cerca, es inversamente proporcional al espacio recorrido.

1.1 Mecanismo general de reemplazo de páginas

Como podemos apreciar en la figura el mecanismo de la memoria virtual es básicamente el mismo que vimos en la paginación, con la única diferencia de que se tiene que obtener en la página de tablas, además de todos los datos que vimos, una bandera que indique si esa página está en memoria o está en disco duro.

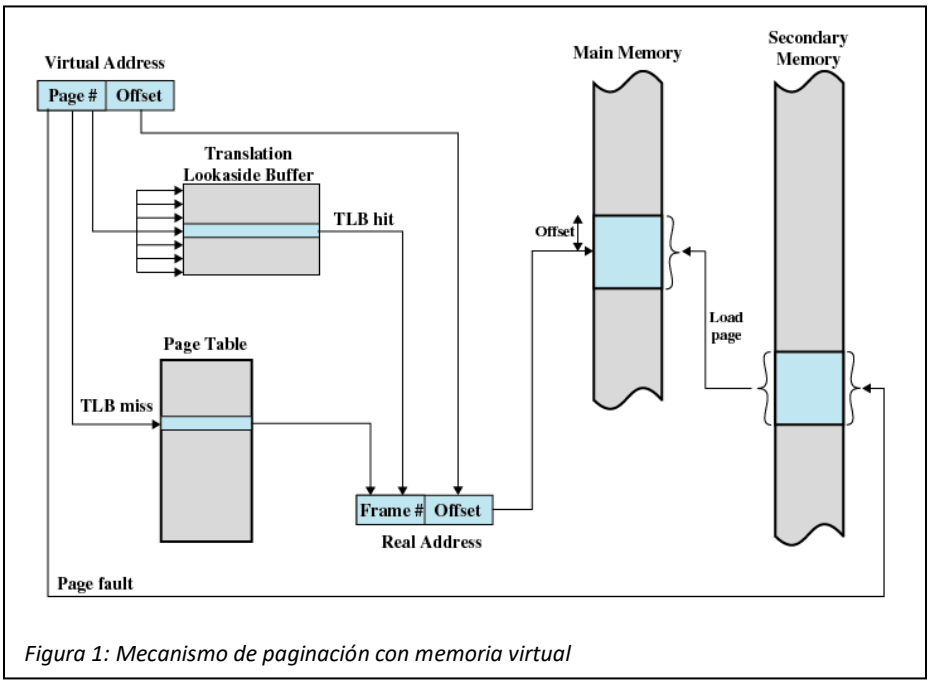


Figura 1: Mecanismo de paginación con memoria virtual

En caso de que la bandera indique que está en memoria el proceso continúa tal como lo vimos en la unidad anterior con un acceso directo a memoria. De lo contrario se invoca una interrupción al sistema operativo indicando un fallo de página, lo cual significa que la memoria que se está intentando acceder no se encuentra en memoria y está en algún lugar del disco duro, por lo que el sistema operativo se debe proceder así:

1. buscar un lugar en la memoria principal, es decir un marco de página que no se esté utilizando y colocar allí la página que está en disco duro.

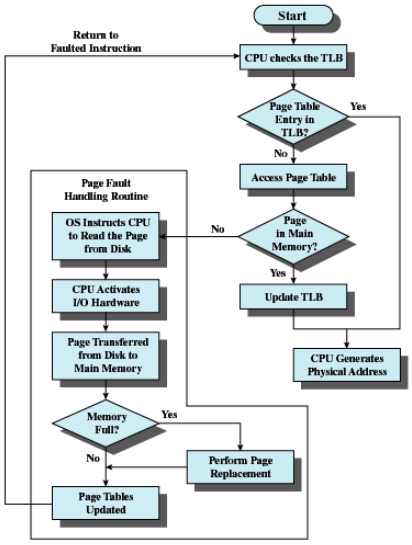


Figura 2: Flujograma del proceso de reemplazo de página

2. Puede suceder que en el área de los marcos de página no existan ningún marco libre donde pueda colocar la memoria que se debe de traer del disco, en cuyo caso se debe elegir una de las páginas que está en memoria, bajo alguna **política de reemplazo**, y bajarla disco duro para dar lugar a la página que se está intentando acceder y provocó el fallo de página.
3. Una vez reestablecida la página, se actualiza la página de tablas con la bandera de que está en memoria y la dirección en memoria donde se cargó.
4. Se reinicia el proceso de acceder a la dirección virtual sólo que esta vez ya estamos seguros de que esa página está en memoria por lo cual se hará por el proceso normal de paginación.

Esto se ilustra en el flujograma de la Figura 2, en la cual se ilustra la lógica que sigue el MMU y el sistema operativo para atender un fallo de página.

1.2 Componentes del manejo de la memoria virtual

Como vimos, hay varios elementos estratégicos involucrados en el manejo de la memoria virtual por parte del sistema operativo estos son:

1. **Estrategia de reposición:** Es la política para decidir qué página en RAM es elegida para bajar a disco, cuando sea necesario subir una página de disco. Es decir, se produjo un fallo de página y es necesario hacer un intercambio.
2. **Estrategia de búsqueda:** Es la política que decide cuándo subir a RAM, páginas que están en disco.
3. **Estrategia de alcance:** Es la política que indica el grupo de páginas que se considerarán al momento de una reposición o intercambio.

2 Estrategias de reposición

Una vez invocado el sistema operativo para solucionar un fallo de página, mencionamos que es posible que en los marcos de página no hay ningún marco libre para poder colocar la página que se necesita desde el disco duro. Por lo tanto, el sistema deberá elegir una de las páginas que están en memoria para descargarla a disco y así hacer el espacio necesario para colocar la página que nos provocó el fallo de página. Cualquier política que se implemente debe buscar alcanzar el principio de optimización:

Principio de optimización: Debe reemplazarse la página que no se va a utilizar durante el tiempo más largo, es decir el que tiene la menor tasa de fallas de páginas de todos los algoritmos.

Por supuesto que este principio es sólo una meta que se busca en cualquier política de reemplazo, ya que es imposible saber el futuro y por lo tanto no podemos predecir cuándo se va a volver a utilizar una página, pero el objetivo final es minimizar la hiperpaginación del procesador para que el rendimiento total del sistema sea aceptable.

Entre las diferentes estrategias de reposición vamos a ver las siguientes:

1. Al azar
2. Primero en entrar primero en salir
3. Segunda oportunidad
4. Menos recientemente usado
5. Menos frecuentemente usado
6. No usado recientemente

2.1 Al azar

Esta política de reemplazo en realidad no es una política utilizada por ningún sistema operativo sino una implementación de referencia para comparar estadísticas con otras estrategias. La implementación de esta política pues resulta bastante simple, aunque en las mismas comparaciones empíricas del rendimiento de una política resulta muy ineficiente en el objetivo de evitar la hiperpaginación.

2.2 PEPS: primero en entrar primero en salir

En este esquema el sistema operativo conserva una lista de las páginas, según cómo se fueron cargando en memoria, ya sea por creación de la página o porque se trasladó del disco duro hacia memoria principal. De esta forma la primera página en la lista es la página que lleva más tiempo en memoria.

La estrategia de PEPS consiste en que al momento de decidir qué páginas se reemplazará, se toma la primera página de la lista, lo que resulta en un esquema fácil de implementar y eficiente en el proceso de decisión. Es decir, en el momento en que el sistema operativo está atendiendo un fallo de página donde necesita decidir, lo más rápidamente posible, la página elegible para descargar a disco.

Sin embargo, este esquema tiene una desventaja qué consiste en que las páginas de uso constante normalmente son enviadas a disco y reemplazadas y restauradas desde disco con mucha frecuencia, porque no se toma en cuenta la cantidad de uso de una página sólo el tiempo en que fue cargada a memoria. Adicionalmente, existe un fenómeno llamado la Anomalía PEPS o anomalía Belady (por su descubridor), que consiste en que, si a un proceso se le asignan más marcos de página para que trabaje, dada una cierta secuencia de accesos a páginas, se producen más fallos de página, lo cual es contrario al sentido común ya que se supone que si a un proceso se le asigna más memoria, debería de producir menos fallos de página.

Como ejercicio considere un proceso que tiene 3 marcos de página para su uso. Si este proceso accede la siguiente secuencia de página: A, B, C, D, A, B, E, A, B, C, D, E. sucederá que, si le agregamos un marco de página más, es esta misma secuencia de páginas, provocará más fallos de página. Es decir, con 3 marcos de página disponible va a producir menos fallos de página a que si tuviera cuatro marcos de página disponibles.

2.3 Segunda oportunidad

Esta estrategia de reemplazo es derivada de la estrategia PEPS con el fin de eliminar el problema de la anomalía Belady. Requiere de una implementación de hardware muy específica que es el bit de referencia (bit R) el cual consiste en que el MMU activa un bit en la entrada de la tabla de páginas cada vez que la página respectiva es utilizada.

Tal como se aprecia en la Figura 3, esta estrategia consiste en una lista circular de páginas referenciadas. Se tiene un apuntador vagabundo hacia la

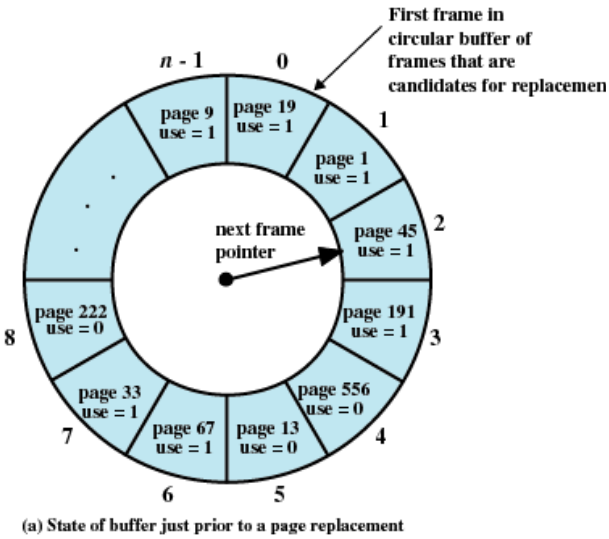
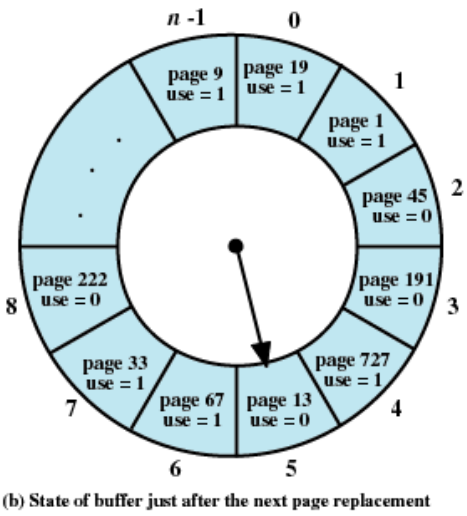


Figura 3: Reemplazo por segunda oportunidad

siguiente página que le corresponde según el orden PEPS, el cual avanza cada vez que se produzca un fallo de página.

Al momento de decidir un reemplazo se analiza la primera de la fila y si tiene el bit de referencia en 1, éste se cambia a 0 y se examina la siguiente página.

Por ejemplo: consideremos Figura 3, la cual representa el estado de la cola circular antes del fallo de página, donde el apuntador a la primera página de la lista está señalando la página 45. Suponiendo que se necesita restaurar del disco la página 727, el sistema analizará la página 45 y determinará que su bit de referencia está activado por lo tanto sólo cambia el bit de referencia a cero y continúan analizando la siguiente página qué es la 191. Esta página también tiene encendido subir de referencia por lo que repite el proceso y analiza la página 556. Dado que esta página tiene su bit de referencia en cero entonces es la elegida para ser reemplazada y en ese marco de página se coloca la página 727 y el apuntador se mueve para la página 13 para estar listo para el próximo fallo de página.



(b) State of buffer just after the next page replacement

Figura 4: Estado de la cola después del reemplazo

Como podemos apreciar esta política es similar a PEPS en simplicidad y eficiencia, Además de eliminar el problema de la anomalía Belady.

Sin embargo, para su implementación se requiere de la capacidad del hardware para activar el bit de referencia de cada página cada vez que se utiliza.

2.4 Menos recientemente usado

En esta política el hardware ayuda en poner una marca de tiempo (la hora) la entrada de la tabla de páginas cada vez que la página correspondiente es accedida. De esta forma, la estrategia consiste en buscar cuál de todas las páginas tiene la menor marca de tiempo es decir la página que no ha sido accedida por más tiempo. Por concepto de localidad temporal es la que menos probabilidades tiene de ser utilizada en el futuro.

La desventaja de este esquema es que cada vez que necesite realizar un reemplazo, se necesita revisar todas las páginas para decidir la que se reemplazará, por lo cual se deben de

recurrir a optimizaciones como tener una lista ordenada por tiempo de todas las páginas, de forma que al momento de decidir se toma la primera de la lista. Esto implica que el mantenimiento ordenado de la lista debe de realizarse en un proceso de fondo para no interferir en el rendimiento del proceso de acceso a las páginas.

Otra desventaja de este esquema es que las páginas de configuración o de datos globales de un proceso suelen ser las primeras en ser cargada esa memoria y ser accedidas a lo largo del proceso, lo que provoca que estas páginas sean descargadas a disco para luego ser vueltas a restaurar, provocando un acceso innecesario a disco. También sucede que algunas páginas pueden ser accedidas sólo una vez y van a permanecer un largo tiempo en memoria hasta que sean las más antiguas de la lista. Para contrarrestar esto se utiliza se utiliza un proceso que periódicamente inserta un bit a la izquierda de cada marca de tiempo (se hace un shift a la derecha con cero en cada marca de tiempo) lo que tiene el efecto de dividir entre 2 el valor de la entrada de páginas y si una página es accedida una vez el valor de su marca de tiempo disminuirá rápidamente produciendo un envejecimiento prematuro que la hará elegible más pronto para un reemplazo.

2.5 Menos frecuentemente usado

En esta política interesa la intensidad de uso de una página, es decir que en vez de mantener una marca de tiempo en la entrada de la tabla de páginas se mantiene un contador de acceso de esta forma se reemplaza la página cuyo contador de acceso tiene el menor valor.

Este contador inicia en cero cuando se crea la página o cuando la página es cargada desde el disco.

Al igual que la política de menos recientemente utilizado, elegir a la página qué se reemplazará requiere recorrer toda la lista de páginas para determinar la que tiene el menor valor. Igualmente se puede aplicar la misma optimización de mantener la lista de páginas ordenada ascendentemente por frecuencia de uso de forma que al momento de elegir se tome la primera de la lista.

2.5.1 Envejecimiento

Se tiene la limitante de que existe un número definido de bits en los contadores lo cual al momento de llegar al máximo valor no se le podrá agregar más, lo que conlleva al problema de una memoria eterna, es decir, si una página fue utilizada mucho, posiblemente al inicio del proceso, pero ya no se volvió a utilizar se conservará bastante tiempo en memoria y posiblemente nunca se descargue a disco. Pero resolver este problema se deben utilizar técnicas de **envejecimiento,** que consisten en qué procesos de fondo reducen el valor del contador periódicamente de forma que, si ya no se utiliza su contador llegue a ser cero y por lo tanto será una candidata para ser reemplazada y bajada a disco.

La más simple de estas técnicas es un proceso que reste periódicamente uno al valor del contador de todas las páginas. Esta es la técnica que utiliza Linux.

Sin embargo, si consideramos que la operación de resta necesita varias microoperaciones para ejecutarse, se puede considerar otra técnica que necesita una operación *shift*, la cual necesita mucho menos microoperaciones que una operación aritmética. Para esta técnica se necesita:

- Que el hardware del MMU tenga integrado el bit de referencia (bit R), el cual se enciende cada vez que se accede una página.
- Un proceso de fondo que reinicia el bit de referencia a 0 periódicamente para todas las páginas.
- Un proceso de fondo insertar a la izquierda del contador el valor del bit de referencia (SHR o desplazamiento a la derecha), de forma que si el bit está encendido se le inserta a la izquierda del contador un 1, en caso contrario se insertará un 0.

De esta forma una página que fue anteriormente utilizada con mucha frecuencia eventualmente llegara a tener el valor cero, dado que periódicamente se le inserto un cero a la izquierda (bit de referencia).

2.6 No usado recientemente

Los esquemas anteriores del menos frecuentemente usado y menos recientemente usado tienen la desventaja de qué se debe mantener una lista de las páginas y se debe hacer un recorrido completo de dicha lista o mantenerla ordenada con el correspondiente costo en rendimiento para poder determinar la página reemplazar. Este esquema busca eliminar ese problema con el uso de 2 bits de hardware:

- Bit de lectura (bit R): se activa cada vez que la página es leída
- Bit de modificación (bit M): se activa cada vez que la página es modificada por el proceso

Adicionalmente también se cuenta con que el bit R es reiniciado periódicamente para todas las páginas. De esta forma se forman cuatro clases de páginas según estén activos los R y M, que se enumeran en la siguiente tabla:

R	M	Descripción
0	0	páginas no leídas ni modificada
1	0	página leída y no modificada
0	1	página no leída y modificada
1	1	página leída y modificada

Notamos que la clase 01 es posible debido a que no es necesario leer una página para modificarla o el bit R fue reiniciado en algún momento. y el bit M se activa si y solo si se está modificando una página.

La combinación de los valores del bit R y M se toman como los dígitos de un numero binario que corresponderían a los valores 0, 1, 2 y 3, por lo que, en esta estrategia, recorre las páginas y reemplaza la primera que encuentre de menor valor, es decir una página de clase 00. aunque en el peor caso, si no hubiera páginas de clase 00 tendría que recorrer todas las páginas para elegir una de las de menor clase, en promedio sólo se recorrerá la cuarta parte de las páginas para elegir una página de clase 00.

2.6.1 RM vs MR

Tener en cuenta que en la literatura existen variantes respecto a la precedencia a los bits R y M. Por un lado, tenemos la estrategia explicada en la cual toma precedencia el bit R, es decir: se prefiere reemplazar una página que no ha sido leída, aunque haya sido modificada, antes que una página leída y no modificada. Por otro lado, se tiene el enfoque contrario, en el cual se prefiere conservar una página en memoria si ha sido modificada. Esta última estrategia toma en cuenta que el reemplazo de una página que ha sido modificada implica escribir en disco dicha página, lo que involucra una operación de disco, mientras que reemplazar una página que no ha sido modificada no implica escribir en disco, si ya existe una imagen de la página en disco, por lo que puede ser más eficiente.

2.6.2 Bit R: READ / REFERENCE

Por otro lado, la funcionalidad del bit R pueden ser implementada de dos formas:

- El bit R se enciende en la lectura: implica si escribimos a una página, no se enciende. Se encenderá el bit **sólo** cuando se lee la página.
- El bit R se enciende en referencia: implica que el bit se encenderá en cualquier referencia, tanto en lectura como en escritura.

3 Estrategias de búsquedas

La otra estrategia que forma parte de la administración de la memoria virtual es la estrategia de búsqueda que define cuándo se debe de restaurar una página de disco hacia la memoria, Y básicamente existen 2:

- Por demanda
- Anticipada

3.1 Por demanda

La búsqueda por demanda es la más simple, ya que se busca una página cada vez que se produce un fallo de página, pero no utiliza ninguno de los conceptos de localización.

Idealmente el sistema operativo debería de buscar sincronizar el tamaño de página con el tamaño de bloque de disco óptimo, pero esto no siempre es posible, por lo que leer página por página no será óptimo.

> **tamaño de bloques de disco óptimo**: es la cantidad de datos X tal que leer del disco una cantidad menor que x se tarda el mismo tiempo y a partir de dicha cantidad el tiempo de lectura/escritura es proporcional a la cantidad de datos leída.

Por ejemplo: leer un byte del disco nos llevará un tiempo X, y leer 100 bytes también nos llevará el mismo tiempo, al igual que leer 32 k. Sin embargo, a partir de 33Kb el disco duro tardará 2X, de tiempo entonces nuestro tamaño de bloque de disco óptimo es de 32 k.

3.2 Anticipada

Se sube a memoria un conjunto de páginas que el proceso muy probablemente utilizará

> **Conjunto de trabajo:** conjunto de páginas que un proceso accedió en un determinado período de tiempo (los que tengan encendido el bit R)

- Se examinan las X referencias a páginas más recientes de un proceso, las cuales componen su conjunto de trabajo
- Implementación por medio del bit R, interrupciones periódicas que limpien este bit. De esta forma las páginas de un proceso que estén con R=1 son las que pertenecen al conjunto de trabajo.
- Se trata de mantener en memoria el conjunto de trabajo de todos los procesos:

- Al inicio, se ponen en los marcos de página la memoria que el proceso solicite dentro de ciertos límites predeterminados
- Al suspender un proceso, por cualquier razón, se conserva la información de cuál es su conjunto de trabajo, y sus páginas se bajan a disco.
- Al despertarse el proceso, se sube a memoria las páginas que conforman su conjunto de trabajo.

4 Estrategia de alcance:

Define la forma en que se asignarán marcos de páginas a los procesos, y la definición del conjunto de páginas candidatas a intercambio, cuando se necesite. Se definen dos posibles estrategias de alcance:

- Alcance local
- Alcance global

4.1 Alcance local

Se asigna a cada proceso un conjunto de marcos de páginas, en el que trabajará. Cuando un proceso produce un fallo de página, la página a ser reemplazada es elegida entre las páginas en los marcos asignados al proceso.

Ventajas: Sólo se tiene que buscar en la tabla de páginas de un proceso
Desventaja: Un proceso muy intenso en memoria, provocará una tasa alta de fallos de páginas, por lo incurrirá en hiperpaginación.

El número de páginas a asignar a cada proceso (m marcos y n procesos) puede ser de dos formas:

- Uniforme: a cada proceso se le asignan m/n marcos de página
- Proporcional: a cada proceso se le asignan marcos de página en proporción al tamaño de la imagen del ejecutable

4.2 Alcance global

Los procesos no tienen asignados un número determinado de marcos de página, y al momento de provocar un fallo, el sistema puede elegir reemplazar una página que pertenece a un proceso diferente al que provocó el fallo de página.

Puede provocar que la hiperpaginación de un proceso, lleve a otros procesos a elevar su tasa de fallos de página.

5 Control de la hiperpaginación

5.1 Por conjunto de trabajo

- Si se tienen el tamaño del conjunto de trabajo CJ_i para un proceso i, entonces

  Demanda= $\sum_i CJ_i$.

- Si la Demanda llega a ser mayor que el número de páginas disponibles, entonces se producirá hiperpaginación, por lo que se elige un proceso y se suspenden, y sus marcos de páginas son asignadas a otros procesos

5.2 Frecuencia de fallas de páginas

- Se establecen límites inferior y superior para la tasa de fallas de páginas
- Si la tasa de fallas de página de un proceso es menor del límite inferior, entonces el proceso tiene muchos marcos asignados, por lo que se le quita uno y se asigna al proceso con mayor tasa de fallas de página
- Si la tasa de fallas de página es mayor del límite superior, entonces el proceso tiene pocos marcos asignados, por lo que se le da un marco más, a costa del proceso con menor tasa de fallos de página. Si no hay marcos disponibles, entonces el proceso se suspende.