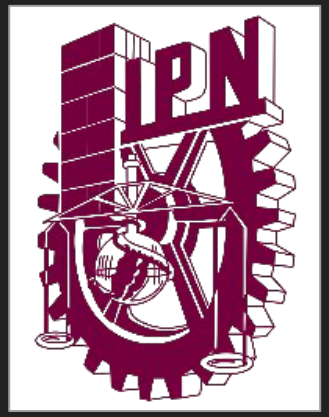


INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

SISTEMAS OPERATIVOS

Unidad 1
Actividad 3

INTEGRANTES DEL EQUIPO:

COLIN RAMIRO JOEL
HERNÁNDEZ REYES JULIO CÉSAR
MALDONADO CERÓN CARLOS
MENDOZA GARCÍA ELIÚ EDUARDO

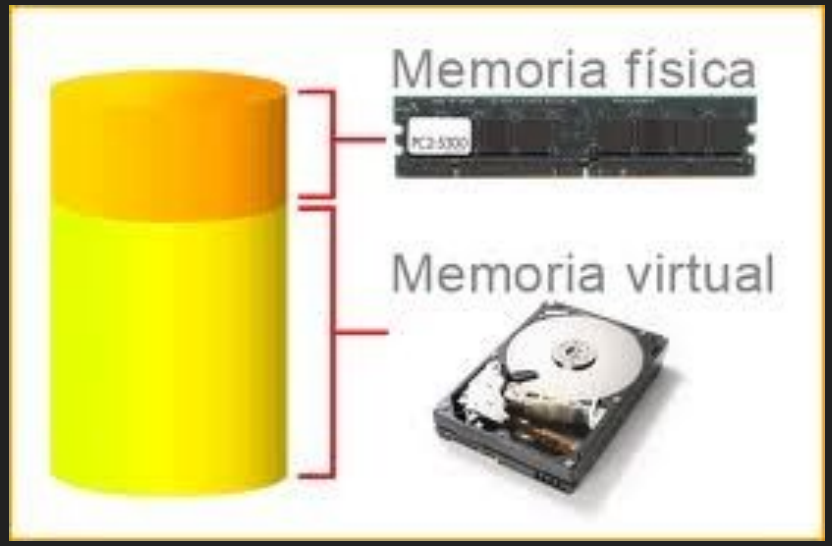
GRUPO: 4CM1

PROFESOR: CORTÉS GALICIA JORGE

Memoria Virtual

Se trata de la memoria que le permite al software usar más de la memoria principal de la que posee una computadora.

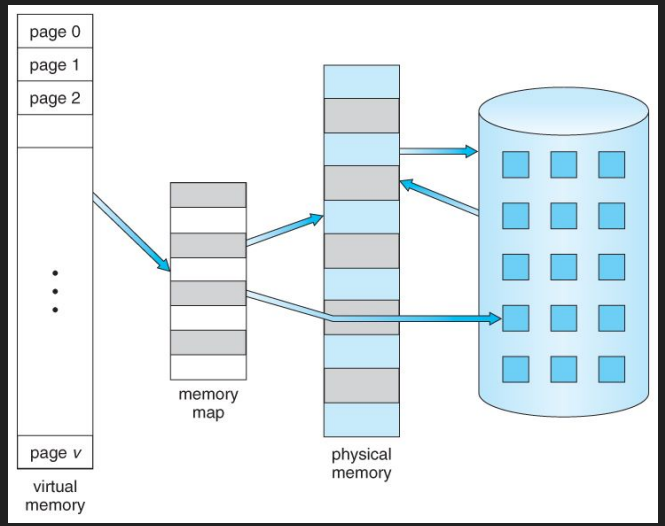
Existen 4 tipos de memoria que una computadora puede poseer :



- ***Cpu***
- ***Memoria Caché***
- ***Memoria Ram***
- ***Disco Duro***

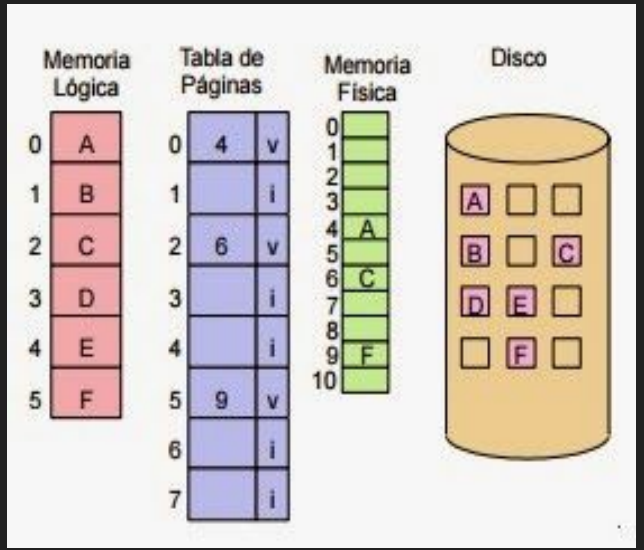
Podemos destacar algunas cosas sobre la memoria virtual y esas son:

- Incluye la separación de la memoria lógica.
- Esta separación permite proporcionar a los programadores una memoria virtual mucho más grande.
- Facilita la tarea de la programación, ya que dichos programadores no deben preocuparse por la cantidad de la memoria física disponible

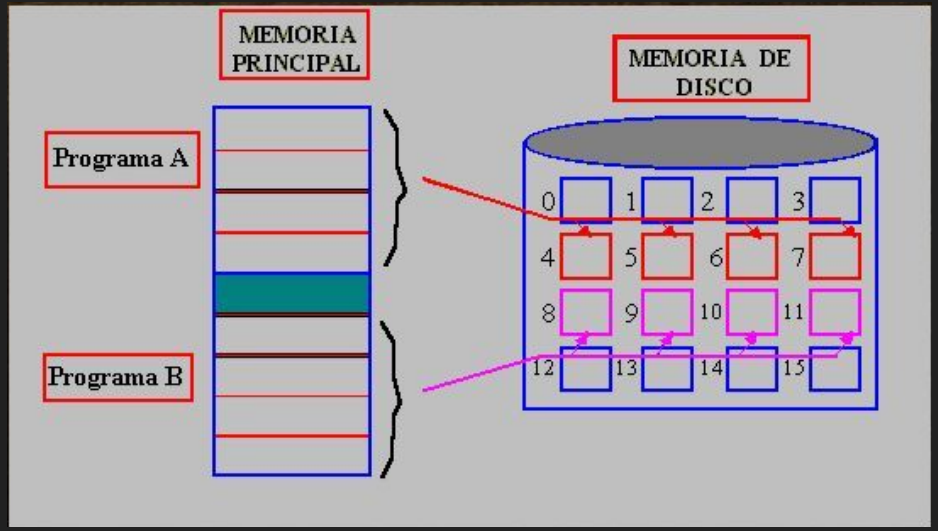


Paginación

- La paginación es un esquema de gestión de memoria el cual permite que el espacio de direcciones físicas de un proceso no sea contiguo.
- Evita el problema de encajar fragmentos de memoria de tamaño variable en el almacén de respaldo.
- Debido a todas las ventajas que contiene esta técnica, la mayoría de los sistemas operativos, utilizan mecanismos de paginación diversos.
- Tradicionalmente el soporte para la paginación se gestionaba mediante el hardware, no obstante, algunos diseños recientes implementan los mecanismos de paginación integrando tanto el hardware como el SO.



Paginación bajo demanda



Este tipo de paginación es similar al sistema de paginación por intercambio, donde los procesos residen en la memoria secundaria (disco).

Solo que en este sistema bajo demanda, en lugar de “intercambiar” todo el proceso a la memoria principal, solo se intercambian las páginas que sean necesarias. Esto se logra con lo que se llama un **INTERCAMBIADOR PEREZOSO**

Rendimiento de la Paginación bajo demanda

En la diapositiva anterior, revisamos el concepto y algunas ventajas de este tipo de paginación, sin embargo, no todo de la paginación bajo demanda es positivo ya que puede afectar significativamente al rendimiento de un sistema informático.

Si se mide o calcula el TIEMPO DE ACCESO EFECTIVO, podremos observar que mientras no se tenga fallo alguno esta medición será equivalente al tiempo de acceso en memoria, no obstante si por alguna razón se produce un fallo de página, se debe leer primero la página relevante del disco y posteriormente acceder a dicha página. Por lo que podemos concluir que no tiene margen de error este tipo Paginación

Hit ratio –% de tiempo la pagina buscada

Hit ratio = α

Effective Access Time (EAT)

EAT = (ciclo_cpu + ε) α + (2*ciclo_cpu + ε)(1 – α)

Sustitución de páginas

Cuando se está ejecutando un proceso de usuario, se produce un fallo de página. El sistema operativo determina dónde reside la página deseada dentro del disco y entonces se encuentra con que no hay ningún marco libre en la lista de marcos libres; tods la memoria está siendo utilizada; esto es la sobreasignación de memoria.

Para solucionar esta cuestión la solución más común es **la sustitución de páginas.**

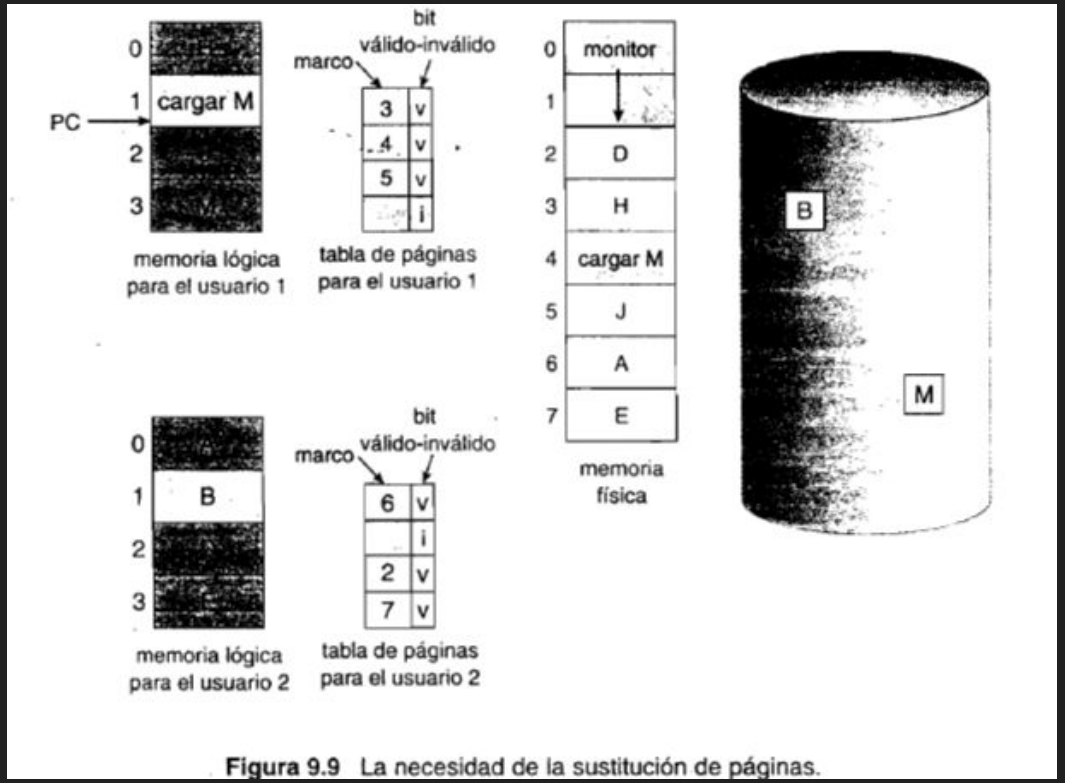


Figura 9.9 La necesidad de la sustitución de páginas.

Sustitución básica de páginas

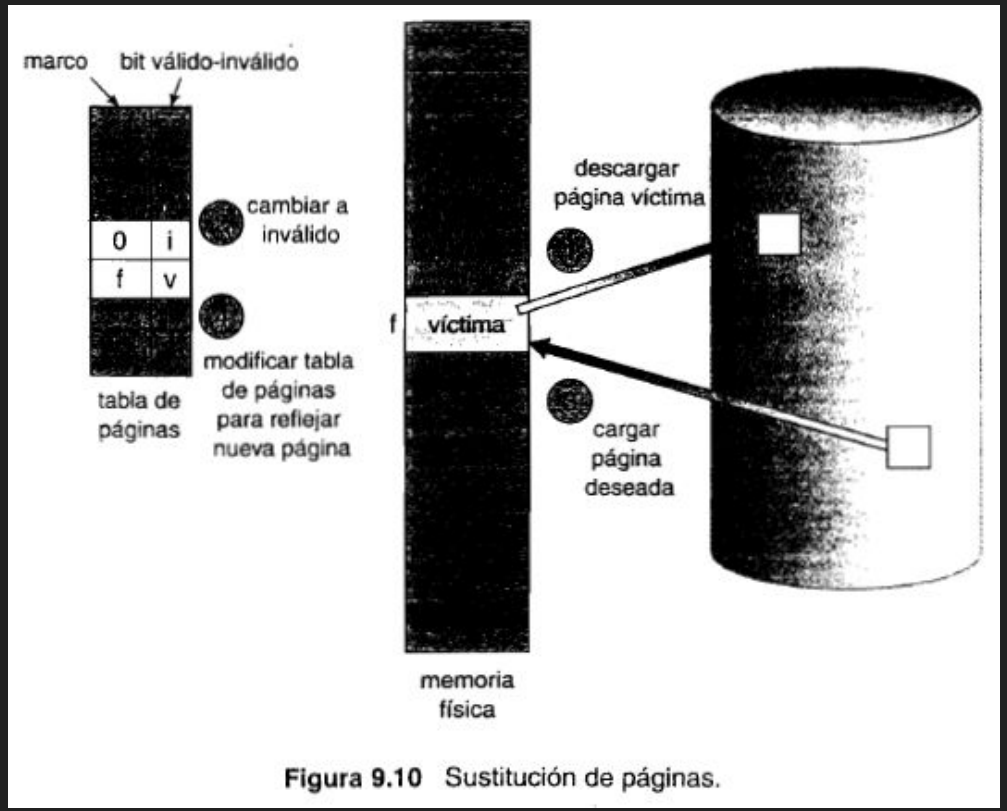


Figura 9.10 Sustitución de páginas.

La sustitución de páginas usa la siguiente técnica. Si no hay ningún marco libre, localizamos uno que no esté siendo actualmente utilizado y lo liberamos. Podemos liberar un marco escribiendo su contenido en el espacio de intercambio y modificando la tabla de páginas para indicar que esa página ya no se encuentra en memoria

Modificando la rutina de servicio del fallo de página para incluir este mecanismo de sustitución de páginas:

1. Hallar la ubicación de la página deseada dentro del disco
2. Localizar un marco libre:
    a. Si hay un marco libre, utilizarlo.
    b. Si no hay ningún marco libre, utilizar un algoritmo de sustitución de páginas para seleccionar un marco víctima.
    c. Escribir el marco víctima en el disco; cambiar las tablas de páginas y de marcos correspondiente.
3. Leer la página deseada y cargarla en el marco recién liberado; cambiar las tablas de páginas y de marcos
4. Reiniciar el proceso de usuario

Sustitución de páginas FIFO

El algoritmo más simple de sustitución de páginas es un algoritmo de tipo FIFO(First-in,First-out). Este algoritmo asocia con cada página el instante en que dicha página fue cargada en memoria. Cuando hace falta sustituir una págin, se elige la página más antigua.

Sustitución óptima de páginas

Este algoritmo sustituye la página que no vaya a ser utilizada durante el período de tiempo más largo. La utilización de este algoritmo de sustitución de página garantiza la tasa de fallos de páginas más baja posible para un número fijo de marcos.

Sustitución de página LRU

El algoritmo LRU(least-recently-used), utiliza el pasado reciente como aproximación del futuro próximo, podemos entonces sustituir la página que no haya sido utilizada durante el período más largo de tiempo.

Sustitución de páginas basada en contador

Al mantener un contador del número de referencias que se hayan hecho a cada página y desarrollar los siguientes dos esquemas:

El algoritmo de sustitución de páginas LFU: requiere sustituir la página que tenga el valor más pequeño de contador.

El algoritmo de sustitución de páginas MFU: se basa en el argumento de que la página que tenga el valor de contador más pequeño acaba probablemente de ser cargada en memoria y todavía tiene que ser utilizada

Algoritmo de búfer de páginas

Algunas versiones del sistema UNIX utilizan este método en conjunción con el algoritmo de segunda oportunidad y dicho método puede ser una extensión útil de cualquier algoritmo de sustitución de páginas.

Aplicaciones y sustitución de páginas

Una base de datos, Almacenes de datos con lecturas de disco secuenciales masivas, etc. Aunque ciertas aplicaciones son más eficientes a la hora de implementar sus propios servicios de almacenamiento de propósito especial, es mejor utilizar los servicios normales del sistema de archivos.

Asignación de marcos

¿Cómo asignar la cantidad fija de memoria libre existente a los distintos procesos?

El caso más simple es de los **sistemas monousuario**. Consideremos un sistema monousuario con 128KB de memoria compuesta de páginas de 1KB. El sistema tendrá 128 marcos. El SO ocupa 35KB, dejando 93 para el proceso de usuario, que se colocan inicialmente en la lista de marcos libres, con una paginación bajo demanda pura. Cuando un proceso de usuario comienza su ejecución, generaría una secuencia de fallos de página. Los primeros 93 obtendrían marcos extraídos de la lista de marcos libres, al agotarse estos, con un algoritmo de sustitución de páginas se seleccionaría una de las 93 páginas en memoria para sustituirla por la página 94, y sucesivamente. Al terminar el proceso, los 93 marcos vuelven a la lista de marcos libres.**También son posibles otras variantes, pero la estrategia básica está clara: al proceso de usuario se le asignan todos los marcos libres.**

Número mínimo de marcos

El número mínimo de marcos está definido por la arquitectura informática.

Peor caso

Se da en aquellas arquitecturas informáticas que permiten múltiples niveles de indirección, resultando en que toda la memoria virtual debería estar en memoria física.

¿Cómo resolverlo?

Debemos imponer un límite en el número de niveles de indirección. Por ejemplo, limitar cada instrucción a un máximo de 16 niveles de indirección. Al producirse la primera, se asigna a un contador el valor de 16, después, el contador se decrementa para cada indirección sucesiva que se encuentre en esta instrucción. Si llega al 0, se produce una interrupción. Esta limitación reduce el número máximo de referencias a memoria por cada instrucción a 17, requiriendo un número igual de marcos.

El número máximo de marcos se define por la cantidad de memoria física disponible.

Algoritmos de asignación

- Asignación equitativa
  La forma más fácil, consiste en dar a cada proceso un número igual de marcos.
- Asignación proporcional
  Se asigna la memoria disponible a cada proceso de acuerdo con el tamaño de éste.

Con ambos mecanismos, a los procesos se les trata igual, independientemente de su prioridad. Sin embargo, puede que nos convenga mejor lo contrario. Una solución consiste en un esquema de asignación proporcional en que el cociente de marcos no depende del tamaño relativo de los procesos, sino de sus prioridades, o bien una combinación de tamaño y prioridad.

Asignación global y local

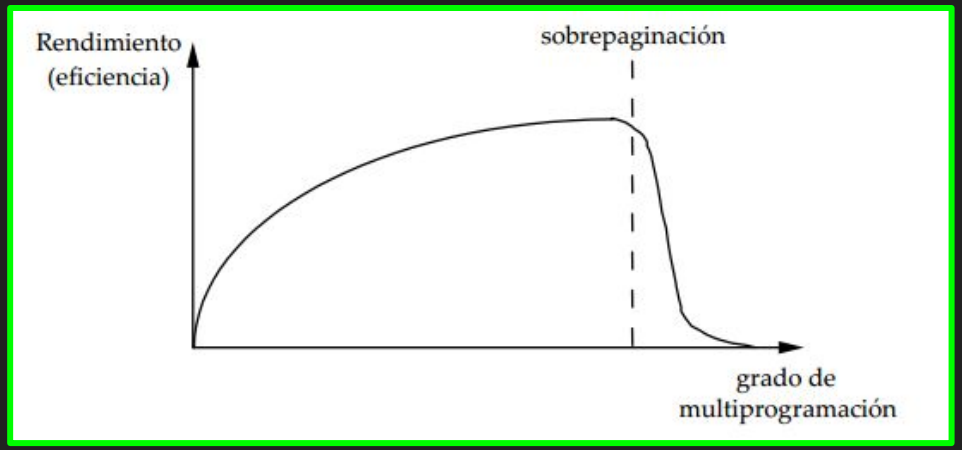
Global:
Permite a un proceso seleccionar un marco de sustitución de entre el conjunto de todos los marcos, incluso un proceso puede quitar un marco a otro.

Local:
Requiere que cada proceso sólo efectúe esa selección entre su propio conjunto de marcos asignados

Si bien ambas presentan posibles problemas en ciertos casos, el mecanismo de sustitución global da generalmente una mayor tasa de procesamiento del sistema, por lo que es el método más utilizado.

Sobrepaginación

Si el número de marcos asignados a un proceso de baja prioridad cae por debajo del número mínimo requerido por la arquitectura de la máquina, deberemos suspender la ejecución de dicho proceso y a continuación descargar de memoria todas sus restantes páginas, liberando así todos los marcos que tuviera asignados.



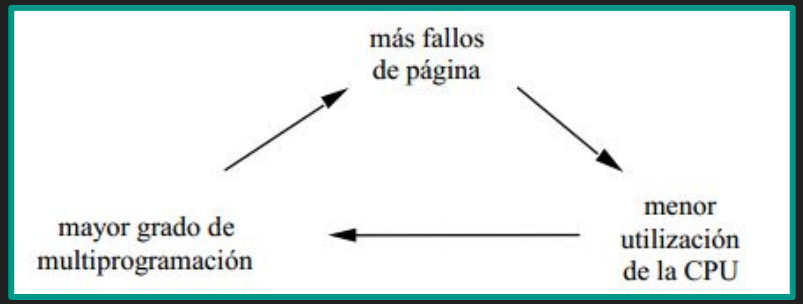
Causa de la sobrepaginación

La sobrepaginación provoca graves problemas de rendimiento.

Podemos limitar los efectos de la sobre paginación utilizando un algoritmo de sustitución local, (o un algoritmo de sustitución basado en prioridades). Con la sustitución local, si uno de los procesos entra en sobrepaginación, no puede robar marcos de otro proceso y hacer que este también entre en sobrepaginación. Sin embargo, esto no resuelve el problema completamente.

Para prevenir la sobrepaginación, debemos proporcionar a los procesos tantos marcos como necesiten.

El modelo de localidad afirma que, a medida que un proceso se ejecuta, se va desplazando de una localidad a otra.



Modelo del conjunto de trabajo

La idea consiste en examinar las A referencias de páginas recientes. El conjunto de páginas en las A referencias de páginas más recientes es el conjunto de trabajo.

Esta estrategia del conjunto de trabajo impide la sobrepaginación al mismo tiempo que mantiene el grado de multiprogramación con el mayor valor posible. De este modo, se optimiza la tasa de utilización de la CPU. La dificultad inherente al modelo del conjunto de trabajo es la de controlar cuál es el conjunto de trabajo de cada proceso.

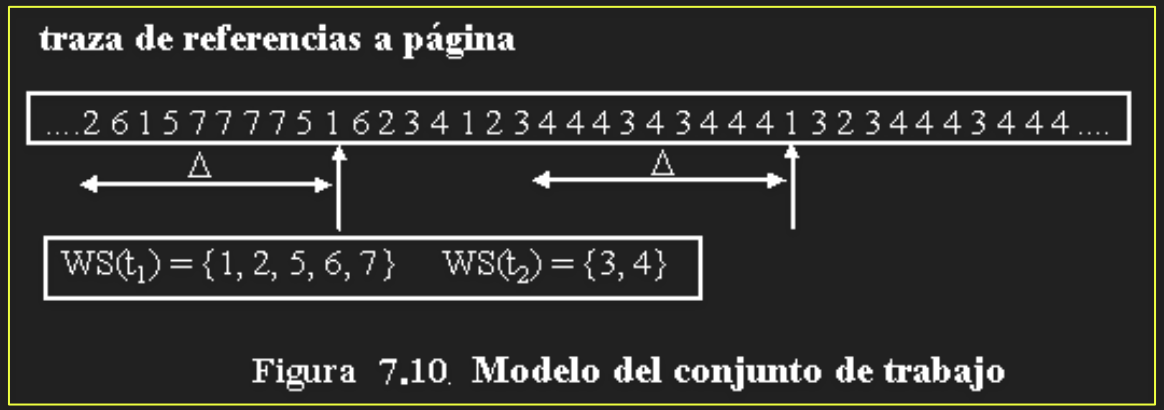


Figura 7.10. **Modelo del conjunto de trabajo**

Frecuencia de fallos de página

El modelo de conjunto de trabajo resulta muy adecuado y el conocimiento de ese conjunto de trabajo puede resultar útil para la sobrepaginación, pero parece una forma un tanto torpe de controlar la sobrepaginación. Hay una estrategia más directa que está basada en la frecuencia de fallos de página.

El problema específico es cómo prevenir la sobrepaginación. Cuando se entra en sobrepaginación se produce una alta tasa de fallos de página; por tanto, lo que queremos es controlar esa alta tasa.