



Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria de
Ingeniería y Ciencias Sociales y Administrativas

Sistemas Operativos

Unidad III: CPU y Administración de memoria

Memoria virtual

M. en C. Hermes Francisco Montes Casiano
hermes.upiicsa@gmail.com



Objetivos

Objetivos

- 1 Comprender la necesidad de contar con memoria virtual
- 2 Entender la diferencia entre la memoria física y la memoria virtual.
- 3 Entender y observar el trabajo en conjunto de los esquemas de administración de memoria física y memoria virtual.



Índice

- 1 Introducción
- 2 Paginación bajo demanda
- 3 Redimiento
- 4 Reemplazo de páginas
- 5 Hiperpaginación





Introducción

- En un sistema que emplea paginación, un proceso no conoce su dirección de memoria relativa a otros procesos.
- Los procesos trabajan con idealización de la memoria, en la cual el proceso cree que ocupan completamente el espacio de direccionamiento, sin importar el tamaño de la memoria.
- En el modelo de paginación los distintos procesos pueden compartir regiones de memoria y direccionar más memoria de la físicamente disponible.
- Por lo anterior, es necesario contar con mecanismos y estrategias que permitan direccionar más memoria de la físicamente disponible.



Introducción

Memoria virtual

Para ofrecer a los procesos mayor espacio en memoria de con el que se cuenta físicamente, el sistema emplea espacio de almacenamiento secundario (disco duro), mediante un esquema de intercambio (swap) guardando y extrayendo páginas enteras.

Gestión de la memoria virtual

La memoria virtual gestionada de forma *automática y transparente* por el sistema operativo.

Los procesos y la memoria virtual

Si un proceso tuviera acceso a directamente a los mecanismos de intercambio **no existiría la memoria virtual.**



Introducción

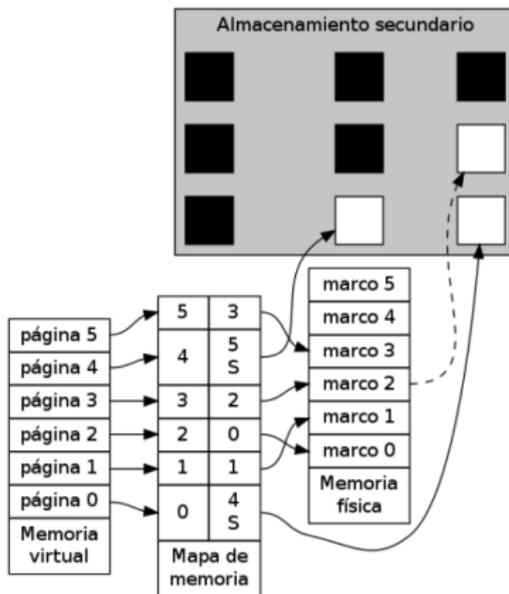


Figura: Esquema general de la memoria



Introducción

- Al utilizar memoria virtual, la memoria física se convierte en una *proyección parcial* de la memoria lógica, la memoria virtual es potencialmente mucho mayor que la memoria física.
- Cuando se habla de memoria virtual, no se está haciendo referencia a el *intercambiador (swapper)*, sino al *paginador*.



Índice

- 1 Introducción
- 2 Paginación bajo demanda
- 3 Redimiento
- 4 Reemplazo de páginas
- 5 Hiperpaginación





Paginación bajo demanda

Introducción

- La memoria virtual entra en acción desde el momento en que los procesos se cargan.
- A una computadora le sería imposible ejecutar código que no esté cargado en memoria.
- Sin embargo, un proceso puede comenzar a ejecutarse sin que esté completamente cargado en memoria; sólo es necesario que se encuentre en memoria la página que le permita al proceso realizar su ejecución actual.

Paginación bajo demanda

Significa que para comenzar a ejecutar un proceso el sistema operativo carga solamente la porción necesaria para comenzar su ejecución (una página o ninguna) y a lo largo de su ejecución el *paginador* cargará las páginas cuando se van a utilizar.



Paginación bajo demanda

Paginador peresozo

- Al emplear un paginador peresozo, las páginas que o serán requeridas nunca se cargaran en memoria.
- La estructura empleada por la MMU para implementar un paginador peresozo es parecida a la de un buffer de traducción adelantada.
- La tabla de paginas incluirá un *bit de validez*, indicando para cada página del proceso si está presente o no en memoria.
- Si un proceso intenta utilizar una página marcada como inválida se provocará un fallo de página, lo que provoca que el sistema operativo suspenda al proceso para traer la página solicitada a la memoria para luego continuar con su ejecución.



Paginación bajo demanda

Fallo de página

- Los pasos a seguir cuando se produce un fallo de página son:
 - 1 Verificar en el PCB si la solicitud corresponde a una página previamente asignada el proceso.
 - 2 Si la referencia es inválida termina el proceso.
 - 3 Se trae la página del disco a la memoria.
 - 4 Solicita al disco la lectura de la página en cuestión hacia el marco especificado.
 - 5 Modifica tanto el PCB como al TLB para indicar que la tabla está en memoria.
 - 6 Termina la suspensión del proceso y continua con la instrucción que provocó el fallo de página.



Fallo de página

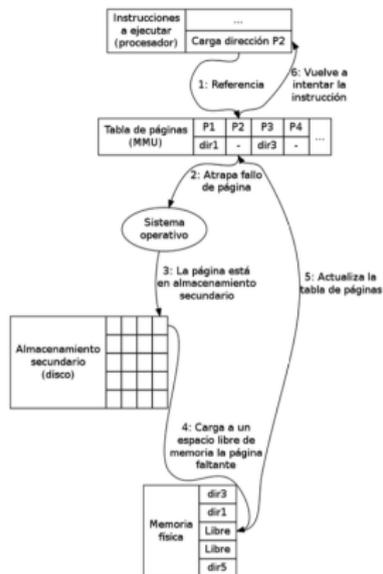


Figura: Fallo de página



Paginación bajo demanda

Paginación bajo demanda pura

- En un sistema de paginación bajo demanda puro ninguna página llegará al espacio de un proceso si no es mediante un fallo de página.
- Un proceso al iniciarse comienza sin ninguna página en memoria y con el registro PC apuntando a una dirección que no está en la memoria.
- El sistema operativo responde cargando la primer página, y con forme avanza el flujo del programa, el proceso irá ocupando el espacio real que empleará.



Índice

- 1 Introducción
- 2 Paginación bajo demanda
- 3 Redimiento**
- 4 Reemplazo de páginas
- 5 Hiperpaginación





Rendimiento

- La paginación bajo demanda puede impactar fuertemente el rendimiento de un proceso, debido al frecuente acceso a disco que el mecanismo requiere.
- Es posible calcular el tiempo de acceso efectivo a la memoria (t_e) a partir de la probabilidad que en un proceso se presente un fallo de página ($0 \leq p \leq 1$), conociendo el tiempo de acceso a la memoria (t_a) y el tiempo que toma atender un fallo de página (t_f).

$$t_e = (1 - p)t_a + pt_f \quad (1)$$



Rendimiento

- Actualmente:
 - $10 \leq t_a \leq 200ns$.
 - $t_f = 8ms$ (5ms posicionamiento de cabezales + 3ms de latencia media)
- Con los valores presentados, en el peor de los casos para t_a y considerando que sólo un acceso a memoria de cada 1000 provoca un fallo de página ($p = \frac{1}{1000}$).

$$t_e = \left(1 - \frac{1}{1000}\right) \cdot 200ns + \frac{1}{1000} \cdot 8000000ns \quad (2)$$

$$t_e = 199,8ns + 8000ns = 8199,8ns \quad (3)$$



Rendimiento

- Con ese promedio se tiene un tiempo de acceso efectivo a la memoria *40* veces mayor que si no se utilizara la paginación bajo demanda.
- Para reducir la degradación del rendimiento, con estos mismos valores, por debajo del 10%, se debe reducir la probabilidad del fallo de página a $\frac{1}{399990}$





Rendimiento

- El hecho de que un proceso incurra en un considerable número de fallos de página no significa que afecte negativamente a todo el sistema.
- La paginación bajo demanda permite que haya más procesos ejecutándose en el mismo espacio de direcciones, **incrementando el grado de multiprogramación.**
- Si un proceso se ve obligado a esperar 8 ms a que se resuelva un fallo de página, durante ese tiempo se siguen ejecutándose los demás procesos.



Rendimiento

Acomodo de las páginas en disco

- Si el espacio asignado a la memoria virtual es compartido con los archivos en disco, el rendimiento se verá afectado.
- Una de las deficiencias de los sistemas de la familia *Windows* es que el espacio de almacenamiento se asigna en el espacio libre del sistema de archivos.

Fragmentación del disco

En los sistemas Windows debido a la fragmentación del disco la memoria virtual queda esparcida por todo el disco duro.

Área de intercambio

En los sistemas Unix se reserva una partición del disco *exclusivamente* para paginación.



Índice

- 1 Introducción
- 2 Paginación bajo demanda
- 3 Redimiento
- 4 Reemplazo de páginas
- 5 Hiperpaginación





Reemplazo de páginas

- Al incrementar el grado de programación con las características del paginación se corre el riesgo de sobre-comprometer la memoria, debido a que los procesos pueden demandar más memoria de la físicamente disponible.

Definición

El reemplazo de páginas es una parte fundamental de la paginación, ya que es la pieza que posibilita una verdadera separación entre memoria lógica y física.

- El mecanismo básico a ejecutar es simple: si todos los marcos están ocupados, el sistema deberá encontrar una página que pueda liberar y guardarla en el espacio de intercambio en disco.
- Una vez liberado el marco se puede utilizar para cargar la página requerida.



Reemplazo de páginas

- Lo anterior requiere una doble transferencia del disco: uno para guardar la página liberada y otro para traer la página de reemplazo.
- Se puede reducir esta doble transferencia, agregando un *bit de modificación* en la tabla de páginas.
- Este bit estará apagado cuando se carga una página y el hardware lo encenderá cuando detecto un acceso de escritura en dicha página.
- Cuando el sistema operativo elige una *página víctima*, si su bit de modificación está encendido implica que dicha página deberá guardarse en el disco. Pero sí está apagado, quiere decir que la información en disco es idéntica a su copia en memoria.



Reemplazo de páginas

Algoritmo

- El encargado de decidir qué página se va a reemplazar marcándola como víctima, es el algoritmo de reemplazo de páginas.
- Lo que busca éste algoritmo es que para un patrón de accesos dado permita obtener el menor número de fallos de página.
- Para analizar un algoritmo de reemplazo, si se busca la cantidad de fallos de página producidos, además de la cadena de referencia, es necesario conocer la cantidad de páginas y marcos del sistema que se está modelando.



Reemplazo de páginas

Primero en entrar primero en salir

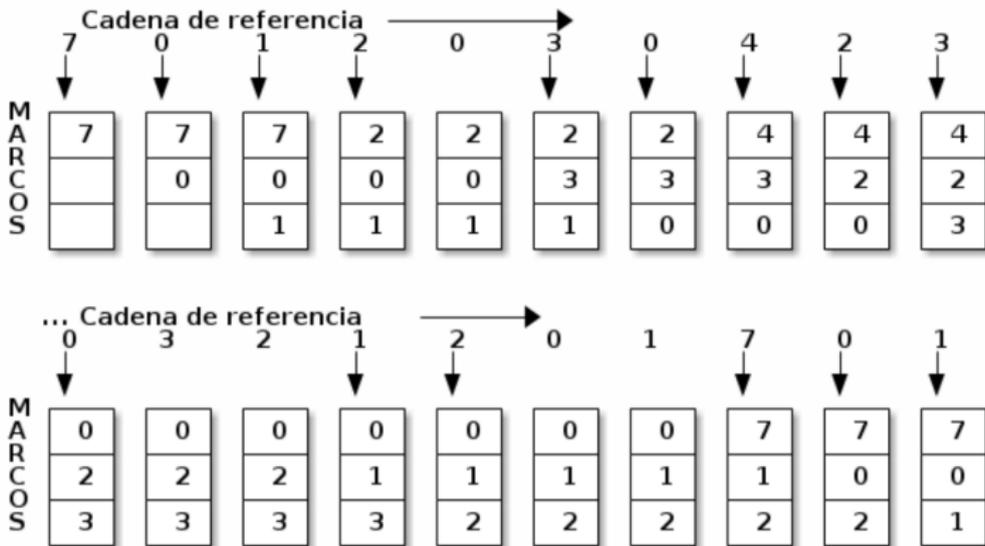


Figura: Algoritmo FIFO de reemplazo de páginas.



Reemplazo de páginas

Reemplazo de páginas óptimo

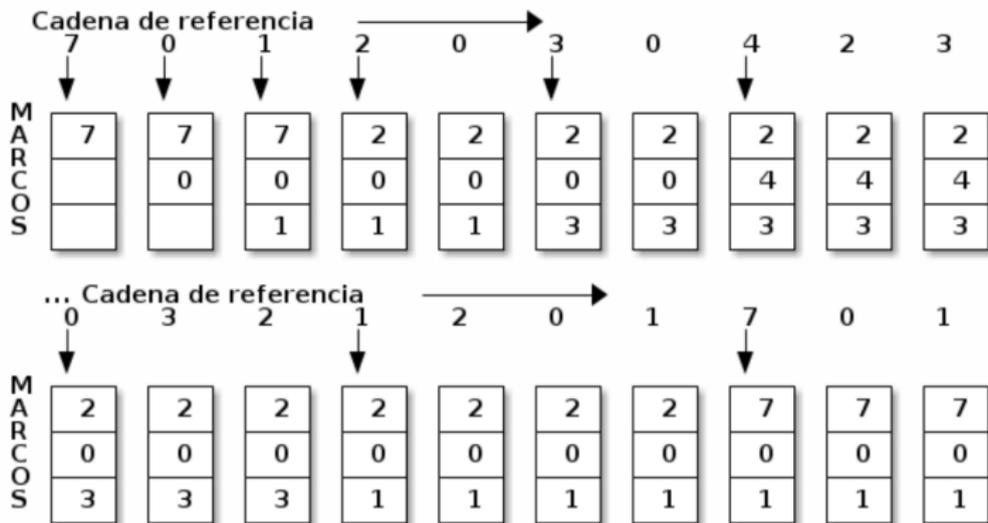


Figura: Algoritmo óptimo de reemplazo de páginas.



Reemplazo de páginas

Menos recientemente utilizado

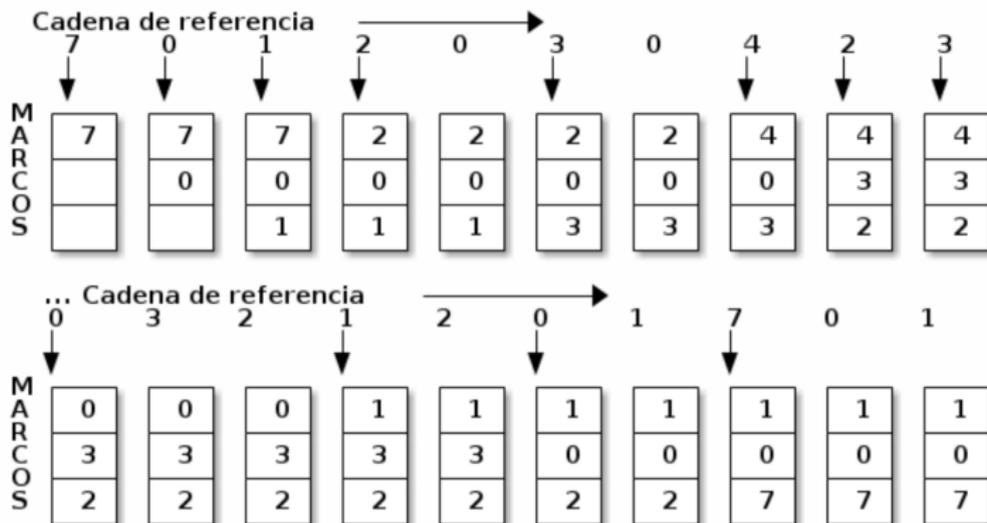


Figura: Algoritmo de reemplazo de páginas menos recientemente utilizadas



Reemplazo de páginas

Reemplazo local

- Busca mantener estable tanto como sea posible el cálculo hecho por es esquema de asignación utilizado.
- Cuando se presenta un fallo de página, las páginas que se consideran candidatas sólo son las páginas del proceso que cause el fallo de página.
- Un proceso tiene asignando su espacio de memoria física y se mantendrá de la misma manera en tanto el sistema operativo no tome otra decisión.



Reemplazo de páginas

Reemplazo global

- Los algoritmos de asignación determinan el espacio de memoria de un proceso al ser inicializados e influyen en los algoritmos de reemplazo.
- Los algoritmos de reemplazo de páginas operan sobre el espacio completo de memoria.
- La asignación física de cada proceso puede variar según el estado del sistema momento a momento.



Reemplazo de páginas

Reemplazo global con prioridad

- Es un esquema mixto, en el que un proceso puede sobrepasar su límite siempre que le *robe* espacio en memoria física a procesos con prioridad inferior.
- Esta política es congruente con el comportamiento de los planificadores.



Reemplazo de páginas

- El reemplazo local es más rígido y no permite aprovechar los periodos de inactividad de algunos de los procesos para mejorar el rendimiento.
- Los esquemas basados en reemplazo global pueden llevar a rendimiento inconsistente: dado que la asignación de memoria física sale del control de cada proceso puede que la misma sección de código presente tiempos de ejecución muy distintos si porciones importantes de su memoria fueron paginadas a disco.



Índice

- 1 Introducción
- 2 Paginación bajo demanda
- 3 Redimiento
- 4 Reemplazo de páginas
- 5 Hiperpaginación





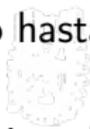
Hiperpaginación

- Es un fenómeno que se puede presentar por varias razones:
 - 1 Cuando un proceso tiene asignadas pocas páginas para realizar su trabajo y genera tantos fallos de página a una frecuencia que le impide realizar trabajo real.
 - 2 Cuando hay en ejecución demasiados procesos en el sistema y los constantes fallos y reemplazos impiden a todos los procesos involucrados avanzar.



Hiperpaginación

- Una salida sería reducir el nivel de multiprogramación
- El sistema puede seleccionar uno (o más) de los procesos y suspenderlos por completo hasta que el sistema vuelva a un estado normal.
- Podría seleccionarse, por ejemplo, al proceso con menor prioridad, al que esté causando más cantidad de fallos o al que esté ocupando más memoria.





Hiperpaginación

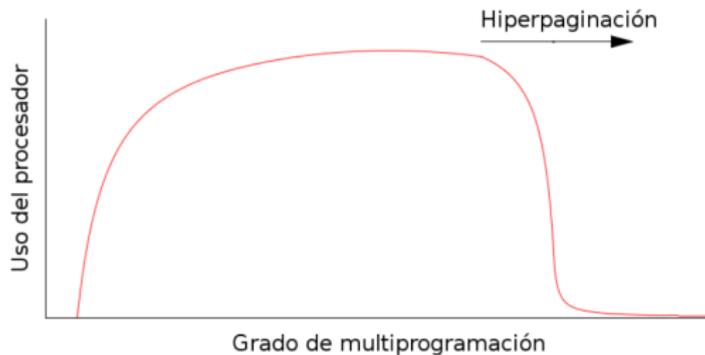


Figura: Caída en el uso del CPU a causa de la Hiperpaginación