

PRÁCTICO 7

Objetivos

- Familiarizarse con las funciones de manejo de memoria de un sistema operativo

Ejercicio 1 (básico)

La siguiente secuencia de requerimientos es solicitada al administrador de memoria:

```
reservar el bloque  $b_1$  de largo 100
reservar el bloque  $b_2$  de largo 500
reservar el bloque  $b_3$  de largo 60
reservar el bloque  $b_4$  de largo 100
liberar el bloque  $b_1$ 
liberar el bloque  $b_3$ 
reservar el bloque  $b_5$  de largo 50
reservar el bloque  $b_6$  de largo 90
```

El largo total de memoria es de 1024. Indicar, para las políticas de asignación de memoria *First Fit* y *Best Fit*, la dirección de comienzo y los largos de todos los espacios libres que queden luego de realizados todos los requerimientos.

Ejercicio 2 (básico)

Dadas particiones de memoria de 100K, 500K, 200K, 300K y 600K (en orden):

- ¿Cómo se colocarían procesos de 212K, 417K, 112K, y 426K (en orden) en memoria empleando los algoritmos de First Fit, Best Fit y Worst Fit?
- ¿Cuál algoritmo aprovecha la memoria de forma más eficiente?

Ejercicio 3 (medio)

Definir las estructuras de datos y escribir los algoritmos necesarios para una administración de memoria por segmentación para las políticas de asignación *First Fit*, *Best Fit* y *Worst Fit*.

Ejercicio 4 (en clase)

En un sistema de memoria virtual por paginado con largo de página 512 bytes y 16 marcos de memoria real, se carga una matriz de 64x64 de tipo double (8 bytes), por filas. Asumiendo una política FIFO de reemplazo de página, cuantos "page fault" ocurrirán al procesar secuencialmente toda la matriz:

- Por filas.
- Por columnas.

Hacer el mismo cálculo para la política LRU.

Ejercicio 5 (avanzado)

Definir las estructuras de datos, escribir los algoritmos y describir las acciones de hardware necesarios para una administración de memoria por paginación para las políticas de reemplazo FIFO, LRU y NUR (algoritmo de segunda oportunidad mejorado).

Ejercicio 6 (básico)

Explique la diferencia entre fragmentación externa e interna.

Ejercicio 7 (básico)

- (a) ¿Qué es el espacio virtual de direccionamiento de un proceso (virtual address space)?
- (b) ¿Este espacio es compartido con otros procesos?
- (c) El sistema operativo por lo general le da cierta estructura a este espacio virtual ¿Cuáles son las componentes básicas?
- (d) ¿Los hilos (threads) que pertenecen a un mismo proceso tienen diferente espacio virtual?

Ejercicio 8 (básico)

¿Qué beneficios brinda un sistema que implemente memoria virtual?

Ejercicio 9 (básico)

En un sistema que maneje memoria virtual ¿Cuál componente se encarga de realizar la traducción de direcciones de virtual a física? ¿Es un componente de hardware o software? ¿Qué utiliza este componente para realizar la traducción en forma más eficiente?

Ejercicio 10 (básico)

Describe las tres estructuras de propuestas en el curso para la tabla de páginas.

Ejercicio 11 (básico)

Las tablas de página de varios niveles optimizan el espacio en memoria con respecto a una tabla de un nivel. ¿Cuáles son las desventajas?

Ejercicio 12 (básico)

¿Cuáles son las ventajas de la segmentación frente a la paginación?

Ejercicio 13 (medio)

En un sistema con paginación:

- ¿Por qué un proceso no puede acceder a memoria que no es de su propiedad?
- ¿Cómo podría el sistema operativo permitir el acceso a otra memoria?
- ¿Por qué debería o no debería hacerlo?

Ejercicio 14 (medio)

Describe el mecanismo por el cual un segmento podría pertenecer al espacio de direcciones de dos procesos distintos.

Ejercicio 15 (medio)

¿Qué entiende por hiperpaginación? ¿Cómo funciona el modelo del working-set?

Ejercicio 16 (medio)

Un sistema operativo usa un sistema de memoria virtual paginada con direcciones virtuales de 32 bits y con páginas de 4KiB. La tabla de páginas reside completamente en memoria principal, siendo el tamaño del descriptor de cada entrada de la tabla de páginas de 32 bits.

El espacio virtual es único por proceso, y las direcciones a partir de la dirección virtual 0 se utilizan para almacenar el área de código más el área de datos. El resto de las direcciones se destinan al espacio de pila, que se almacena comenzando en la dirección virtual más alta y creciendo hacia las direcciones bajas.

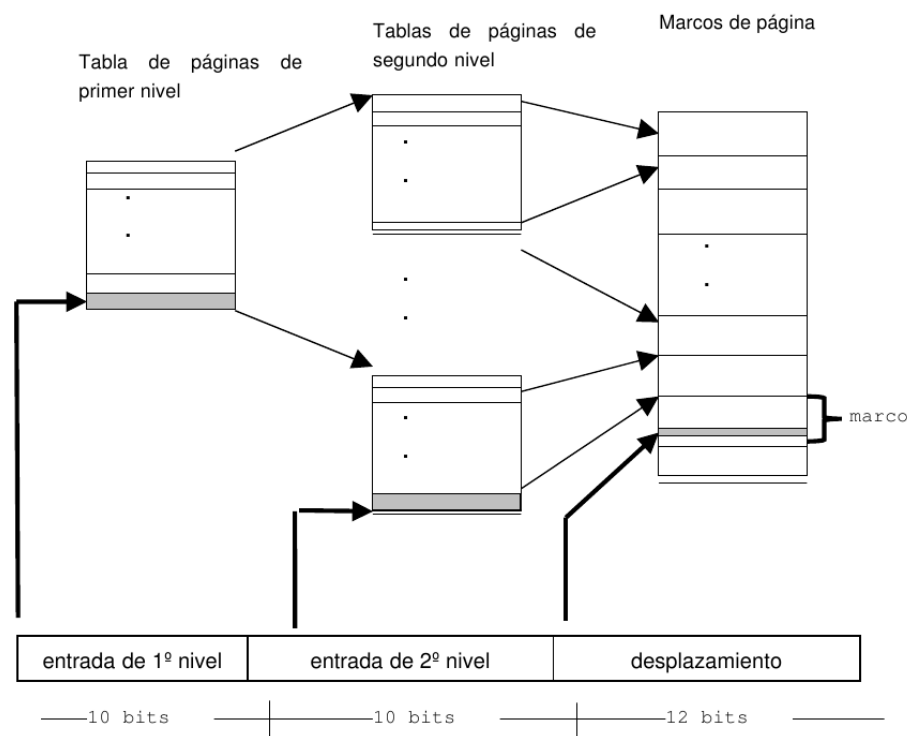
Asumimos que tenemos un proceso que nunca llega a necesitar más de 16 MiB. De este espacio, 12 MiB se utilizan para almacenar el código y los datos mientras que los 4 MiB restantes se destinan al espacio de pila.

Responda justificando adecuadamente las siguientes preguntas:

(a) ¿Cuánto ocupa la tabla de páginas de cada proceso?

(b) ¿Cuál es la ocupación útil (entradas ocupadas) por nuestro proceso en % de la tabla de páginas?

Posteriormente, se decide usar un sistema de tablas de dos niveles, como se muestra a continuación.



Una dirección virtual se interpreta de la siguiente manera: Los 10 bits más significativos sirven para seleccionar la entrada de la tabla de primer nivel. Esta entrada contiene el número del marco donde reside la tabla de páginas de segundo nivel. Los 10 bits siguientes se usan para seleccionar una entrada de la tabla de segundo nivel. Esta entrada contiene el número del marco donde está cargada la página. Una vez localizado el marco, solo hace falta sumarle los 12 bits del desplazamiento de la dirección virtual.

Suponiendo de nuevo el mismo proceso que ocupa como máximo 16 MiB, y sabiendo que la tabla de primer nivel siempre está cargada en memoria durante la ejecución del proceso, mientras que las tablas de segundo nivel se cargan bajo demanda, responda justificando adecuadamente:

(c) ¿Qué tamaño tiene cada una de las tablas?

(d) ¿Cuántas tablas de segundo nivel utiliza el proceso? ¿Qué índices se ocupan de la tabla de primer nivel?

(e) ¿Cuánto espacio de memoria se necesita para almacenar simultáneamente todas las tablas necesarias?