

Sistemas Operativos

Práctico 8

Curso 2025

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:

1. Reservar el bloque b_1 de 100 bytes.
2. Reservar el bloque b_2 de 500 bytes.
3. Reservar el bloque b_3 de 60 bytes.
4. Reservar el bloque b_4 de 100 bytes.
5. Liberar el bloque b_1 .
6. Liberar el bloque b_3 .
7. Reservar el bloque b_5 de 50 bytes.
8. Reservar el bloque b_6 de 90 bytes.

El total de memoria disponible es de 1024 bytes. Inicialmente toda la memoria se encuentra libre. Indicar la dirección de comienzo y el tamaño de todos los espacios libres luego de realizados todos los requerimientos considerando las políticas de asignación de memoria First Fit y Best Fit.

Ejercicio 2 (básico) Dadas particiones de memoria libre de 100 kB, 500 kB, 200 kB, 300 kB y 600 kB (en orden):

- (a) ¿Cómo se colocarían procesos de 212 kB, 417 kB, 112 kB, y 426 kB (en orden) en memoria empleando los algoritmos de First Fit, Best Fit y Worst Fit?
- (b) ¿Cuál algoritmo aprovecha la memoria disponible más eficientemente?

Ejercicio 3 (medio) Escriba el pseudocódigo de los algoritmos de asignación First Fit, Best Fit y Worst Fit. Defina las estructuras de datos necesarias.

Ejercicio 4 (básico) La memoria física de un sistema tiene **256 bytes** divididos en bloques libres:

- **Bloque 1:** Dirección inicial `0x0000`, tamaño 64 bytes
- **Bloque 2:** Dirección inicial `0x0040`, tamaño 32 bytes
- **Bloque 3:** Dirección inicial `0x0080`, tamaño 128 bytes

Tres procesos solicitan memoria:

- **P1:** 32 bytes
 - **P2:** 64 bytes
 - **P3:** 16 bytes
- (a) Para cada estrategia de asignación de memoria (*First Fit*, *Best Fit*, *Worst Fit*), determinar:
 - Dirección base de asignación para cada proceso
 - Bloques restantes después de cada asignación
 - (b) Traducir las siguientes direcciones lógicas a físicas para cada estrategia:
 - P1: `0x0010`

- P2: 0x0020
- P3: 0x0008

Ejercicio 5 (medio) 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 64×64 de tipo double (8 bytes), por filas. Asumiendo una política FIFO de reemplazo de página, calcular cuántos fallos de página ocurrirán al procesar secuencialmente toda la matriz:

1. Por filas.
2. Por columnas.

Realizar el mismo cálculo para la política LRU.

Ejercicio 6 (avanzado) Definir las estructuras de datos, escribir los algoritmos y describir las acciones de hardware necesarias para una administración de memoria por paginación, para las políticas de reemplazo FIFO, LRU y NUR (algoritmo de segunda oportunidad mejorado).

Ejercicio 7 (medio)

- (a) En un sistema con paginación:
 - i. ¿Por qué es imposible para un proceso acceder a memoria que no es de su propiedad?
 - ii. ¿Cómo podría el sistema operativo permitir a un proceso acceder a la memoria de otro proceso?
 - iii. ¿Por qué debería o no debería hacerlo?
- (b) Describa las acciones que realiza la rutina de atención de fallos de página en un sistema con paginación bajo demanda.
- (c)
 - i. ¿Qué entiende por hiperpaginación?
 - ii. ¿Cómo funciona el modelo del working-set?

Ejercicio 8 (avanzado) Se tiene un sistema de memoria con paginación y una TLB (Translation Look-aside Buffer). La tabla de páginas cuenta con 16 entradas. La TLB cuenta con 6 entradas y utiliza un algoritmo de reemplazo FIFO. El tiempo de acceso a la TLB es de 0.001 segundos y el tiempo de acceso a la memoria principal es de 0.1 segundos. Al momento de ejecutar un proceso, su tabla de páginas y la TLB están cargadas como se muestra en la Figura 1.

Suponga que las entradas de la TLB en la imagen están ordenadas por antigüedad. Es decir, la entrada para la página 4 es la más antigua y la entrada para la página 6 es la más reciente.

Se pide:

- (a) Se realiza la siguiente secuencia de acceso a páginas: 7, 8, 9, 0, 1. Brinde un estimativo del tiempo de acceso total para dicha secuencia y el estado resultante de la TLB.
- (b) ¿Qué ocurre si el proceso intenta acceder a la página 14? Describa la secuencia de pasos que debe realizar el sistema.
- (c) Se observa que en la tabla de páginas, algunas secuencias contiguas de páginas se corresponden con secuencias contiguas de frames en memoria. A su vez, se sabe que los accesos regularmente se dan entre secuencias contiguas de páginas (por ejemplo: 3, 4, 10, 11). Explique cómo modificaría el algoritmo de reemplazo de la TLB para mejorar el hit-rate en la realidad planteada.
- (d) Volviendo al algoritmo de reemplazo FIFO original. Suponga que los procesos de su sistema realizan una enorme cantidad de accesos a las páginas que almacenan el código y unos pocos accesos a las páginas que almacenan el stack y la memoria dinámica. También suponga que el código usualmente se almacena entre las páginas 0 y 4. Explique cómo modificaría el algoritmo de reemplazo de la TLB para mejorar el hit-rate en la realidad planteada.

Tabla de Paginas

	Frame	Valido/ Invalido
0	505	V
1	506	V
2	507	V
3	403	V
4	404	V
5	128	V
6	423	V
7	424	V
8	425	V
9	426	V
10	522	V
11	523	V
12	0	I
13	0	I
14	0	I
15	0	I

TLB

Pagina	Frame
4	404
9	426
0	505
7	424
5	128
6	423

Figura 1: Contenido de la tabla de páginas y la TLB