

## **Trabajo Práctico N° 5: Administración de memoria convencional (no virtual) y virtual**

### **Memoria convencional (no virtual)**

1. Dadas particiones de memoria de 100K, 500K, 200K, 300K y 600K (en ese orden). Para los algoritmos first-fit, best-fit y worst-fit:
  - (a) ¿cómo se ubicarían los procesos de 212K, 417K, 112K y 426K (en ese orden)?
  - (b) ¿Qué algoritmo utiliza la memoria más eficientemente?
  - (c) ¿Los resultados se condicen con los estudios teóricos/experimentales existentes?
2. Suponga un hipotético sistema que cuenta con 32K de memoria principal para asignación a procesos de usuario, y un espacio de direccionamiento lógico de 8 páginas por proceso de 1K cada una:
  - (a) ¿Cuál es la cantidad disponible de marcos?
  - (b) ¿Cómo se conforma la dirección física?
  - (c) ¿Cómo se conforma la dirección lógica?
  - (d) ¿Cuál es el tamaño máximo (en KB) que un proceso puede tener?
  - (e) ¿Cuál es el espacio máximo (en KB) ocupado por una tabla de páginas?
3. Considere un sistema con direcciones lógicas y físicas de 32 bits, y páginas de 4Kb. Considere que en la tabla de páginas se almacena si la página es válida o no.
  - (a) ¿Cuánta memoria se puede direccionar?
  - (b) ¿Cuál es el tamaño de la tabla de páginas, si la tabla es de un solo nivel?
  - (c) Proponga un sistema de tablas de páginas de 2 niveles de igual cantidad de entradas y calcule la eficiencia para procesos de: 4Kb, 4Mb, 512Mb y 3Gb.
4. Considere un sistema con direcciones lógicas y físicas de 64 bits, y páginas de 4Kb.
  - (a) ¿Cuánta memoria se puede direccionar?
  - (b) ¿Sería viable implementarlo con tablas de páginas de un solo nivel?
  - (c) Proponga un paginado de más de un nivel que permita utilizar este tipo de direccionamientos.
5. Considere un sistema con paginación con la tabla de páginas almacenada en memoria:
  - (a) Si una referencia a memoria lleva 200 ns ¿Cuánto lleva una referencia a memoria paginada?
  - (b) Si agregamos registros asociativos y un 75% de los accesos a la tabla de páginas se encuentran en dichos registros ¿cuál es el tiempo de referencia a memoria efectivo si el acceso a registro lleva 10ns?
6. ¿En qué casos los espacios de direccionamiento lógico y físico pueden variar? Ejemplifique.

### **Memoria virtual**

1. Una computadora provee un espacio virtual de direcciones de  $2^{32}$  bytes sobre una memoria física de  $2^{18}$  bytes. La memoria virtual es implementada mediante paginado, siendo el tamaño de página es de 4096 bytes. Un proceso genera la dirección virtual 11123456. Explique cómo se establece la dirección física.
2. En una memoria con paginado por demanda se mantiene la tabla de páginas en registros. Demora 8 ms reemplazar una página si hay un frame disponible o si la página reemplazada no fue modificada, y 20 ms si la página reemplazada fue modificada. El tiempo de acceso a memoria es de 100ns. Asumir que la página a ser reemplazada ha sido modificada el 70% de las veces. ¿Cuál es la máxima cantidad aceptable de fallos de página para un tiempo de acceso efectivo no mayor a 200ns?

3. Se tiene un S.O. con paginado por demanda con 3 frames (inicialmente libres). Un proceso efectúa la siguiente secuencia de accesos a memoria:

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1

Determinar el número de fallos de página para cada uno de los siguientes algoritmos:

- (a) FIFO
  - (b) Algoritmo óptimo
  - (c) LRU
  - (d) LFU
4. Utilizando el algoritmo FIFO de reemplazo de páginas, graficar fallos de página en función del número de frames (1 a 7) para la siguiente secuencia de accesos:

1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

Explicar qué sucede cuando se aumenta el número de frames desde 3 a 4. Repetir el problema para LRU.

5. Considere una matriz bidimensional de 100x100 enteros (cada entero ocupa dos bytes). El elemento (1,1) de la matriz se encuentra en la página 1 de un sistema con páginas de 200 bytes. Un proceso situado en la página 0 manipula la matriz. El sistema posee 3 frames, y el proceso que manipula la matriz se encuentra en el frame 1; los otros dos frames están libres. Mediante reemplazo de páginas LRU determine la cantidad de fallos de páginas generados por:

```
for(int j=0; j<100; j++)
    for(int i=0; i<100; i++)
        a[i][j]=0;

for(int i=0; i<100; i++)
    for(int j=0; j<100; j++)
        a[i][j]=0;
```