

**EJERCICIOS TIPO PARCIAL DE MÓDULO 5**

1) Considere un sistema operativo (S.O.) que utiliza paginación bajo demanda, direccionamiento lógico de 20 bits que permite como máximo 512 páginas por proceso, algoritmo de reemplazo de página LRU, sustitución global y cuenta con 4 frames totales para procesos de usuario. Se ejecutan dos procesos correspondientes a un procesador de texto cuyo código ocupa 2.1 KB. Todas las páginas de código se cargan al iniciar el proceso, son compartidas y no producen fallo de página en ese instante. El área de datos del proceso A ocupa 11.5 KB y del proceso B 17 KB.

Durante su ejecución, los procesos realizan las siguientes operaciones:

Proceso	Operación	Página
A	Lectura	Código 1
A	Lectura	Datos 1
A	Escritura	Datos 2
B	Lectura	Código 1
B	Escritura	Datos 1
B	Escritura	Datos 3
A	Lectura	Código 1
A	Escritura	Datos 1
A	Lectura	Datos 3
B	Lectura	Código 1

Se pide: graficar el estado de los frames en cada momento e indicar cuando hay swap-in y cuando swap-out.

2) En una computadora que cuenta con 24 bits para direccionamiento (tanto lógico como físico), opera un Sistema Operativo (S.O.) que administra la memoria mediante paginación bajo demanda y tiene una MFT de 2048 entradas. Existe un proceso, con cuatro frames asignados, que emitió la lectura de las siguientes direcciones lógicas (en decimal); 185528 - 185529 - 185579 - 186479 - 15775265 - 15775765 - 15784291 - 15775794 - 15776291 - 14276291 - 14266791 - 14266785 - 15776293 - 185543 - 6458426 - 6457905 - 15774504 - 14267173 - 14103589. Sabiendo que toda carga de página se considera un fallo y que el proceso no tiene cargada ninguna página en memoria, **se pide; mostrar el estado de los frames, siempre que haya cambios en los mismos, para los métodos FIFO, LRU y LFU, especificando cuando hay swap in y cuando swap out, (en caso de empate de víctimas, se define por FIFO).**

3) Una empresa mantiene la información de sus empleados en tres archivos con formatos diferentes entre sí; norte (1550 registros - 82 Bytes por registro), sur (1176 registros - 78 Bytes por registro) y central (2036 registros - 80 Bytes por registro). Recientemente se decidió unificar (los formatos) y centralizar dicha información en un solo archivo llamado empleados (66 Bytes por registro), para esto se diseñó un programa que opera de la siguiente manera: realiza la carga de la información desde disco a memoria (**carga cada archivo en un array distinto**), luego almacena toda la información recopilada en un **cuarto array**, es decir que transfiere el array de norte por completo, luego el de sur y por último el de central, para posteriormente realizar su volcado a disco en el archivo empleados.

Sabiendo que el sistema donde se corre el programa utiliza paginación bajo demanda, que el tamaño de la memoria es de 4GB, que la máxima cantidad de páginas que se le puede asignar a un solo proceso es 131072, que el programa dispone de 3 frames para cargar páginas de datos, que toda carga de página se considera fallo, que el algoritmo de reemplazo de páginas es FIFO y que el tiempo en cargar una página a memoria es de 10 ms, pero si también hay que hacer swap-out, el tiempo se triplica, se pide:

a. Mostrar el estado de los frames (siempre que haya cambios en los mismos) durante todo el procedimiento.

- b. Calcular el tiempo total que insumirá el intercambio de páginas del programa.
- c. Indicar y explicar si existe alguna manera de reducir el tiempo del punto anterior, sin reescribir el programa.
- d. Calcular el porcentaje de fragmentación existente dentro de la última página utilizada por el vector de empleados.

4) Se tiene un sistema con una memoria real de 16 Kbytes y trabaja con paginación bajo demanda, donde cada página tiene un tamaño de 1024 bytes, donde la asignación de memoria a los procesos es equitativa (cada proceso recibe  $\text{frames\_libres/cantidad\_procesos frames}$ ) y además la sustitución de páginas es local con un algoritmo de reemplazo tipo LIFO. También se sabe que un int ocupa 4 bytes y un char 1 byte. Que el S.O. tiene un tamaño de 4Kbytes y se aloja a partir de la dirección 0000. En cuanto a los procesos las primeras páginas en ser cargadas son las de código. Se tienen los siguientes programas que pasan a ejecutar simultáneamente en dicho sistema:

Programa A: Su código ocupa 513 bytes y debe leer una tabla de una base de datos, que tiene 1000 registros de 63 bytes cada uno, imprimiendo cada registro secuencialmente.

Programa B: Su código ocupa 200 bytes y la parte mas significativa es la siguiente:

```
int vec[512],sum=0;
for(int pos=0;pos<512;pos++)
    sum=sum+vec[pos];
```

**Se pide:**

- a. ¿Cuántos fallos de página producirá cada programa? Considere que toda carga de página produce un fallo.
- b. Grafique el estado final de la memoria, especificando las direcciones (en decimal) de comienzo y fin del área de memoria asignada a cada proceso usuario.
- c. ¿Existe algún programa que genere una cantidad de fallos de página excesiva con respecto a los otros? En caso afirmativo, ¿Qué modificación puede realizarse sobre el sistema para disminuir dicha cantidad de fallos? ¿Por qué?