Sistemas Operativos
2° Parcial 1C2018 - TM - Resolución

<u>Aclaración</u>: La mayoría de las preguntas o ejercicios no tienen una única solución. Por lo tanto, si una solución particular no es similar a la expuesta aquí, no significa necesariamente que la misma sea incorrecta. Ante cualquier duda, consultar con el/la docente del curso.

Teoría

1. **Conjunto Residente**: Es la porción de un proceso que se encuentra en la memoria principal en un momento determinado.

   Tamaño del conjunto Residente: Para esta característica se utilizan dos políticas: Asignarle al proceso una cantidad fija de frames durante su existencia en el sistema o asignarle una cantidad de frames que pueda variar durante la vida del proceso.

   Alcance del reemplazo: Para esta característica se utilizan dos políticas: alcance global, en donde la página a reemplazar se elige de todos los procesos en memoria; y alcance local, en la que la página a reemplazar se elige de entre todas las pertenecientes al proceso que produjo el fallo de página.

	Reemplazo Local	***Reemplazo Global***
Asignacion Fija	La cantidad de frames asignados al proceso es fija. La página a reemplazar es elegida de entre los frames asignados al proceso.	No es posible
Asignacion Variable	La cantidad de frames asignados al proceso se modifica cada cierta cantidad de tiempo. La página a reemplazar es elegida de entre los frames.	La página a reemplazar es elegida de entre todos frames de memoria, sin importar que proceso produjo el fallo de página. Esto produce que el tamaño del Resident Set de los procesos varíe constántemente

2. Falso. La MMU también valida la presencia de la página en la memoria. El sistema operativo participa al momento de evaluar si la página es inválida o si hay que ir a buscarla a disco.
3. No es suficiente. Debe también marcar los bloques de datos como libres, en el bitmap de bloques libres.
4. Ambos algoritmos comparten las siguientes características:
    a. Hacen uso del algoritmo SCAN para planificar.
    b. A la hora de planificar, utilizan colas que no permiten el ingreso de nuevas solicitudes.
    c. No generan inaninción.

    Se diferencian en:
    d. N-STEP-SCAN usa una cantidad indefinida de colas de tamaño N, mientras que F-SCAN solo trabaja con dos colas.

    La principal característica que los diferencia del resto es que ambos algoritmos garantizan la atención de todos lo pedidos y evitan la inanición.
5. V o F
    a. Falso. Puede generar ya que asigna particiones de distintos tamaños (la potencia de 2 más chica en la que entra el proceso).
    b. Falso. Tiene la mínima cantidad de entradas entre lo que puede direccionar y los bloques reales que tiene la partición.

Practica

Ejercicio 1

2 platos = 4 cabezas
Para calcular las direcciones físicas debo dividir los números decimales / Cabezas* Sectores por pistas. Ósea 4 cabezas * 15 sectores por pistas = 60

D1: T=0 ms los cilindros son: 133, 6 T = 25 ms los cilindros son: 65, 3
D2: T=0 ms los cilindros son: 158, 33 T = 1ms los cilindros son: 33,65

a) FSCAN utiliza dos colas C1 y C2

Disco 1 = D1
T = 0 ms C1=133, 6 T= 25 ms C2= 65, 3

Atiendo solo la C1:
D1: 35 /133/ 299 (tope) /6 pasaron (265 cil +293 Cil) * 1 ms = 558 ms

T = 559 ms se abre la C2: 65,3
D1: 6 / 3 / 0(inicio) / 65 pasaron (6cil + 65 cil) * 1 ms = 71 ms

Orden de pedidos: 35 / 133/ 300 (tope) /6 / 3 / 0 (inicio) / 65.
Tiempo total de Búsqueda: 558 ms + 71 ms = 628 ms

Disco 2 = D2
T = 0 ms C1=158, 33 T= 1 ms C2= 33, 65

Atiendo solo la C1:
D2: 35 /158/ 300 (tope) / 33 pasaron (264 cil +266Cil) * 1 ms = 530 ms

T = 531 ms se abre la C2: 33, 65
D2: 33/ 33 /0(inicio) / 65 pasaron (33 cil + 65 cil) * 1ms= 98ms

Orden de pedidos: 35 / 158 / 299 (tope) /33 / 33 / 0 (inicio) / 65.
Tiempo total de Búsqueda: 530 ms + 98 ms = 628 ms

Si existe diferencia en tiempo de búsqueda por el orden en que llegan los pedidos y el disco 2 repite su cilindro.

b) SSTF
Disco 1 = D1
T = 0 ms C=133, 6
D1: 35 /6 pasaron 29 cil * 1 ms = 29ms

T =29ms C: 133, 65, 3
D1: 6 / 3 / 65 /133 pasaron (3 cil + 130 cil)* 1 ms = 133 ms

Orden de pedidos: 35 /6 / 3 / 65 / 133.
Tiempo total de Búsqueda: 29 ms + 133 ms = 162 ms

Disco 2 = D2
T = 0 ms C=158,33
D1: 35 /33 pasaron 2cil * 1 ms = 2ms
T =2ms C: 158,33,65
D1: 33 / 33 / 65 /158 pasaron 125 cil* 1 ms = 127 ms
Orden total:
Disco: 35 /33 /33 / 65 / 158.
Tiempo total de Búsqueda: 2 ms + 127 ms = 127 ms

c) Tiempo de búsqueda es más pequeño para ambos discos aplicando el algoritmo SSTF dado que no se visitan los extremos y por el orden en que van llegando los pedidos que están muy cercanos entre ellos.

Ejercicio 2

a) Se sabe que todos los bloques mostrados pertenecen al archivo. Dentro de los que son de datos, tres de ellos contienen el contenido “hola”. Por lo tanto hay que averiguar cual de ellos fué leído.
Al rastrear las referencias a los tres bloques que contienen el dato, se puede ver que son referenciados por indirecciones simple, doble y triple.
Si se consumieron 8Kb al leer el dato, sería necesario saber el tamaño de bloque, de esa forma se sabría cuántos accesos se tuvieron que realizar, y de ahí la indirección usada.
Al indicarse que la máxima fragmentación interna es de 2047 bytes, se puede concluir que los bloques son de 2Kb, dado que el máximo de fragmentación posible en todo sistema es siempre el tamaño del bloque - 1.
Por lo tanto, 8Kb/2Kb = 4 accesos a disco. Por lo tanto, podemos decir que el bloque leído es el 14, dado que se accede a través de una indirección triple.

b) El tamaño máximo real de un archivo es el mínimo entre el tamaño máximo teórico del mismo y el tamaño real disponible en disco.

Tamaño máximo teórico del archivo:

12 p.d. x 2Kb + 1 p.i x punterosEnBloque x 2Kb + 1 p.i.d x punterosEnBloque ^ 2 x 2Kb + 1 p.i.t x punterosEnBloque ^ 3 x 2Kb

El dato pendiente aquí es el tamaño de puntero. Para ello, sabemos que el tamaño máximo direccionable es 32 Zb = 2^75.

Por lo tanto: 2 ^ tamañoPuntero x 2 ^ 11 bytes (tamaño bloque) = 2^72.
Por lo tanto: x = 64 bits, por ende los punteros son de 8 bytes.
Consecuentemente, la cantidad de punteros por bloque es = 2048 / 8 = 256

Volvemos a la cuenta original del tamaño máximo teórico de un archivo:

12 p.d. x 2Kb + 1 p.i x 256x 2Kb + 1 p.i.d x 256x^2 x 2Kb + 1 p.i.t x 256 x^3 x 2Kb ~= 32Gb

Para el límite de disco, hay que considerar la partición y no el disco completo, dado que el archivo se encuentra limitado al volumen de la partición.

Por lo tanto, el máximo real es Min(32Gb, 16Gb) = 16 Gb.

Ejercicio 3

La dirección lógica es de 16 bits:
Una justificación:
3 bits para las páginas porque las tablas de páginas son de tamaño fijo y de hasta 8 páginas.
1 Bit para el segmento (consideramos que todos los procesos pueden tener máx 2 segmentos).
12 bits (el resto) para desplazamiento.

Otra justificación:
Todas las DF son válidas, la única forma para que los bits de más peso hagan referencia a números de frames válidos, es que se usen 12 bits para el offset.
Con esa data, y viendo que hay procesos con más de 4 páginas, al menos se necesitan 3 bits para cada página. El restante, es el segmento.

La memoria de 64 KiB está dividida en 16 marcos de 4 KiB cada uno.

1-

1. Física 5111h -> lógica 8111h: segmento 1 / página 0 de PID1
2. Física B333h -> lógica 9333h: segmento 1 / página 1 de PID1.
3. Física 8111h -> lógica 8111h: segmento 1 / página 0 de PID2.
4. Física 7444h -> lógica 0444h: segmento 0 / página 0 de PID1.

2- Cómo sólo hay 3 marcos libres en la memoria, es posible utilizando mecanismos para compartir páginas. Creando un nuevo “proceso 3” que comparta las páginas pertenecientes al segmento 0 del proceso 1, ya que nunca se modifican.

PID3 - segmento 0		PID3 - Segmento 1	
# Marco	Protección	# Marco	Protección
7	RX	13	RW
6	RX	15	RW
10	RX	-	-
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	-

3- Máximo teórico de un proceso: dos segmentos de 8 páginas cada uno. Máximo 16 páginas = 64 KiB
Tamaño de la memoria es 64 KiB. Coinciden. Por lo tanto, el Real es 64 KiB.

4- No hay fragmentación externa. El máximo de fragmentación interna es 4095 por segmento. Por lo tanto 8190 bytes.