Sistemas Operativos
2° Parcial 1C2019 - TT - 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. MMU: mejora la eficiencia de la memoria realizando la traducción de Dirección Lógica a Física. Las ventaja principal es permitir que la traducción se realice de manera más rápida. Además, aporta protección al sistema analizando la validez de la dirección lógica.

   TLB: mejora la eficiencia de la memoria para que la traducción sea más rápida. Es una memoria caché de entradas de la tabla de páginas del proceso actual (algunas tienen entradas de varios procesos), La consulta en la TLB es más rápida que en la tabla de páginas. Mejora aún su eficiencia cuando se utiliza paginación jerárquica.

   Disco de swap: permite que los procesos sean más grandes que la memoria o bien cargar más procesos en memoria.

   Existen otras opciones válidas también.

2. V o F
   a. Falso, la ventaja de utilizar paginación jerárquica es que las tablas de páginas ocupan menos espacio en memoria real.
   b. Falso, la paginación no requiere compactación debido a que no sufre fragmentación externa.

3. Un inodo con punteros directos e indirecciones permite manejar tanto archivos de diferentes tamaños, de forma eficiente. Los archivos pequeños serán accedidos usando los directos, y a medida que vayan creciendo se irán accediendo a las indirecciones. Para estos casos, el costo de acceder a bloques extras (los bloques de punteros) es relativamente bajo.

   Respecto a la fragmentación, nunca sufrirá de fragmentación externa, pero sí puede sufrir interna, hasta casi un bloque.

4. En las E/S síncronas, la ejecución del proceso que las utiliza se detendrá hasta que la misma haya finalizado. En las asíncronas, el proceso podrá continuar su ejecución, ya que el resultado de la operación no será necesario inmediatamente. Para recibir este

resultado, se utilizará una función *callback* que será llamada cuando la operación termine. Para el primer caso, el buffer simplemente será llenado y vaciado cuando se realice la operación. Para el segundo, el buffer puede llenarse durante una cantidad de tiempo, y el Sistema Operativo puede vaciarlo cuando lo considere oportuno, mejorando la performance.

5. Una ventaja que plantea PGU es que ocupa bastante poco espacio comparado con la técnica de matriz de acceso que pierde mucho espacio dado que se utiliza una sola tabla para todos los archivos del sistema, aparte si se agrega un nuevo archivo siempre se debe agregar una nueva columna. Lo mismo si se agrega un nuevo usuario.
   Una desventaja de pgu es que no tiene granularidad por usuario, si se quiere ser mas especifico se lo tiene que complementar con una ACL. La matriz de accesos permite especificar para cada usuario el tipo de permiso que tendrá.

Práctica

1) Sabemos que según la configuración, tenemos 160 bloques lógicos por cilindro, con lo cual vamos a calcular en que cilindro se encuentra cada uno de las siguientes solicitudes:

Sector	Cilindro		Sector	Cilindro
2240	14		2444	15
640	4		4640	29
7	0		81	0
80	0			

a) SSTF = 1 + [(7-4) +(4-0)+(0-0)+(0-14)+(14-15)+(15-29)+(29-0)] * 2ms
= 131 ms.

LOOK = 1 + [(7-4) +(4-0)+(0-0)+(0-14)+(14-15)+(15-29)+(29-0)] * 2ms
= 131 ms.

b) FSCAN = 1 + [(7-4) +(4-0)+(0-0)+(0-14)] * 2ms + [(14-15)+(15-29)+(29-99)+(99-0)] * 2ms
= 411 ms.

c) Para todos los casos, el orden de atención de los pedidos es el mismo. Para el punto a) el tiempo de búsqueda es el mismo para los dos algoritmo y menor con respecto a FSCAN. En caso del FSCAN al atender los pedidos utilizado 2 cola de atención y algoritmo SCAN para cada cola, el tiempo de búsqueda es mayor. La ventaja de utilizar el algoritmo FSCAN con respecto a los otros, es que los pedidos no sufren de inanición.

2) Los archivos tienen que soportar 16 GB. Llamaremos al tamaño de bloque "BL". Buscamos aquel bloque con menor desperdicio.

Sabemos entonces qué:

12 BL + BL x BL/4 + BL x (BL/4)^2 + BL x (BL/4)^3 >= 16 GB

También sabemos que el término más significativo en el tamaño del archivo es "BL x (BL/4)^3". Por ende podemos calcular

BL x (BL/4)^3 = 16GB
BL^4 / 2^6 = 2 ^34 → BL^4 = 2^40 → BL = 2^10 → 1 KB

a) Un bloque más chico que 1 KB no permitiría archivos de 16GB. Un bloque más grande generaría fragmentación mayor.
b) 1 KB x 2 ^32 = 2 ^42 → 4 TB.

3) Según las tablas de páginas el estado inicial de la memoria es:

	#marco	Memoria Real	
	0	P1 - Segmento 0 - Página 2	Bit de Uso: 1
Ptr P2----->	1	P2 - Segmento 0 - Página 0	Bit de Uso: 1
	2	P1 - Segmento 1 - Página 2	Bit de Uso: 1
	3	P2 - Segmento 1 - Página 1	Bit de Uso: 1
	4	P2 - Segmento 0 - Página 1	Bit de Uso: 0
	5	P2 - Segmento 1 - Página 2	Bit de Uso: 1
	6	P2 - Segmento 0 - Página 2	Bit de Uso: 1
	7	P1 - Libre	Bit de Uso: -
	8	P2 - Segmento 1 - Página 3	Bit de Uso: 0
	9	P2 - Segmento 2 - Página 3	Bit de Uso: 0
Ptr P1----->	10	P1 - Segmento 0 - Página 3	Bit de Uso: 1
	11	P1 - Segmento 1 - Página 3	Bit de Uso: 0
	12	P1 - Segmento 1 - Página 0	Bit de Uso: 0

13	P2 – Segmento 2 – Página 1	Bit de Uso: 1
14	P2 – Segmento 2 – Página 0	Bit de Uso: 0
15	P2 – Libre	Bit de Uso: -

Las direcciones lógicas tienen 2 bits para el segmento, 2 bits para la página (se obtiene del tamaño de las tablas) y 12 para el offset (se obtiene del tamaño de las páginas).
Las dirección físicas tienen 4 bits para el marco y 12 para el offset.

-P1 3111h: segmento 0 – página 3 está en memoria se traduce como A111h.

-P1 1222h: segmento 0 – página 1 no está en memoria. Se utiliza frame libre 7 que está libre y alguna vez fue asignado al proceso 1. Se traduce como 7222h.

-P1 5333h: segmento 1 – página 1 no está en memoria. Hay Page fault es necesario reemplazar una página. El puntero apunta al frame 10. El siguiente frame con el bit de uso en 0 del proceso 1 es B333h.

-P2 A444h: segmento 2 – página 2 no está en memoria. Se utiliza frame libre 15 que está libre y alguna vez fue asignado al proceso 2. Se traduce como F444h.

-P2 4111h: segmento 1 – página 0 no está en memoria. Hay Page fault es necesario reemplazar una página. El puntero apunta al frame 1. El siguiente frame con el bit de uso en 0 del proceso 2 es 4111h.

-P2 3222h: segmento 0 – página 3 no está en memoria. Hay Page fault es necesario reemplazar una página. El puntero apunta al frame 5. El siguiente frame con el bit de uso en 0 del proceso 2 es 8222h.
Estado final de las tablas:

P1 – Segmento 0

#F	P	U
0	0	1
7	**1**	**1**
0	1	1
10	1	**0**

P1 – Segmento 1

#F	P	U
12	1	0
11	**1**	**1**
2	1	1
11	**0**	**0**

P2 – Segmento 0

#F	P	U
1	1	**0**
4	**0**	**0**
6	**1**	**0**
8	**1**	**1**

P2 – Segmento 1

#F	P	U
4	**1**	**1**
3	1	**0**
5	1	**0**
8	**0**	**0**

P2 – Segmento 2

#F	P	U
14	1	0
13	1	1
15	1	1
9	1	0

Puntero P1 apunta a frame 12.
Puntero P2 apunta a frame 9.