

Sistemas Operativos

2° Parcial 1C2017 - TT - Resolución

Aclaración: 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 profesor del curso.

Teoría:

1) El particionamiento fijo es más sencillo de implementar, ya que utiliza estructuras de tamaño fijo para manejar las particiones libres y ocupadas (ej: tabla de páginas y bitvector). El dinámico requiere de estructuras de tamaño variable para esto mismo (ej: lista con particiones ocupadas y otra con particiones libres). El fijo presenta fragmentación interna, en la última partición que se otorga a cada proceso. El dinámico presenta fragmentación externa, la cual se soluciona usando compactación. Cuanto mayor sean las particiones, mayor será el problema de fragmentación asociado.

2) Dado que se asigna una entrada por frame, no es posible utilizar memoria virtual sin realizar ninguna modificación.

3) RAID 0+1 divide en bandas los datos a escribir en los discos, y luego realiza un espejado de los mismos. RAID 5 divide los discos en bandas, reservando un disco para guardar la paridad de cada banda. Esta paridad se distribuye entre todos los discos, evitando los problemas que ocasionan las escrituras simultáneas sobre esquemas anteriores (como RAID 4). En RAID 0+1 podrían fallar hasta dos discos, siempre y cuando no sean la pareja de espejos. En RAID 5 puede fallar un único disco.

4) El área de swap es una porción en memoria secundaria (habitualmente en un disco), que se utiliza para guardar procesos (o partes de ellos) que no están en memoria principal. De esta forma, es posible suspender procesos o bien implementar mecanismos de memoria virtual. Por lo general se implementa utilizando una partición o bien un archivo con el mismo fin.

Dado que los procesos que se colocan de forma contigua, lo ideal es utilizar asignación contigua.

5) Cuando se utilizan archivos mapeados a memoria, el archivo se comporta como parte de la imagen del proceso. De esta forma, el proceso puede accederlo y modificarlo como a cualquier porción de memoria, sin tener que ocuparse de leer y escribir en el filesystem. Esta operatoria se torna transparente, dado que el Sistema Operativo se ocupa de todo, aprovechando los mecanismos que se utilizan para Memoria Virtual. Es conveniente utilizarlos cuando se quiere aprovechar todo esto que se comentó, y cuando los archivos no

son demasiado grandes, ya que las funciones para mapear archivos suelen tener limitaciones al respecto.

Práctica:

1)

a) La página 1 del proceso 0 debería estar en el frame 1, pero está en el 2. La página 4 del proceso 10 debería estar en el frame 4. El resto está en la posición correcta.

b) 20B00h -> Página 20h -> 32 -> $f(32, 5) = \text{Resto}(32 + 25, 10) = 7$
La dirección física es 7B00h

c) 2CCAh -> Frame 2 -> Pagina 1 del proceso 0
La dirección lógica es 1CCAh

2)

a) $4 \text{ GiB} = 2^{32} \rightarrow 2^{32} / 2^{16} = 2^{16} \rightarrow$ Los bloques serán de 64 KiB

b) Se accederá una vez a la FAT y a dos clusters

c) FAT permite encadenar todos los bloques, si fuese necesario, con lo cual un archivo podría utilizar todos los clusters. En ese caso, el tamaño teórico sería de $2^{16} \times 2^{16} = 2^{32} = 4 \text{ GiB}$. En la práctica, deberíamos descontar el espacio ocupado por la FAT dos veces (uno para la FAT y otro para el duplicado que se guarda por seguridad)

2^{16} entradas \times 16 bits/entrada = $2^{17} = 128 \text{ KiB}$

d) Para tener menor fragmentación interna, necesitaremos clusters más pequeños. Por ende, podríamos tener bloques de 4KB y punteros de 32 bits. Cada bloque de punteros tendrá 1024, lo cual nos dará 4TB en el tercer nivel de indirección, superando ampliamente lo pedido. El problema que tienen estos esquemas es que requieren de accesos extras para los bloques de punteros, aunque estos son pocos comparados con los bloques de datos a los que se busca acceder.

3)

a) 35 -> 37 -> 40 -> 43 -> 56 -> 89 -> 99 -> 15 -> 9 -> 2 \rightarrow TT = 322 ms

b) 35 -> 37 -> 40 -> 43 -> 56 -> 89 -> 99 -> 2 -> 9 -> 15 \rightarrow TT = 356 ms