

2° Parcial 1C2018 - 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/la docente del curso.

Teoría

1. Falso. Si bien las mismas se traducen cuando es necesario, siguen permaneciendo en forma de direcciones lógicas en memoria.
2. Falso. El algoritmo Best fit genera mayor fragmentación externa, por dejar porciones libres más chicas que con el algoritmo Worst fit.
3. Falso. Los algoritmos FIFO, N-STEP-SCAN y F-SCAN nunca generan inanición.
4. Falso. Siempre se puede crear un archivo vacío si hay inodos disponibles y espacio en la entrada del directorio
5. Falso. Por el principio de localidad espacial se pueden cargar páginas contiguas (por ejemplo en de código), por lo que sería conveniente usar asignación contigua para dicha partición de swap para poder realizar una sola lectura en el disco, en lugar de tener que realizar múltiples en distintos bloques y/o sectores.

Practica

Ejercicio 1

4 platos = 8 Cabezas

Disco 4 GB = $4 * 1024 * 1024 = 4294967296$ bytes

2048 sectores por pistas

Sectores de 2 KiB

Cantidad de Cilindros = $\text{Disco} / (\text{Cabezas} * \text{Sectores por pistas} * \text{tamaño sector})$

Cantidad de Cilindros = $4294967296 / (8 * 2048 * 2048) = 128$ Cilindros

a) N-Step-Scan con N=3

T = 0ms Pistas C1: 115, 60 T = 10ms C2: 99

Disco: 50 / 60 / 115 Fin de la C1 pasaron 65 Cil * 1 ms = 65 ms

T = 65 ms Pistas C2: 99, 100, 65 y C3: 83, 125

Disco: 115 / 127 (Tope) / 100 / 99 / 65 fin C2 pasaron 74 Cil * 1 ms = 74 ms

T = 65 ms + 74 ms = 139 ms pistas C3: 83, 125

Disco: 65 / 25 / 0 (inicio) / 83 / 125 pasaron 190 cil * 1 ms = 190 ms

Orden total:

Disco: 50 / 60 / 115 / 127 (tope) / 100 / 99 / 65 / 25 / 0 (inicio) / 83 / 125.

Tiempo total de Búsqueda: 65 ms + 74 ms + 190ms = 329 ms

b) SSTF

T = 0 ms Pistas C: 115, 60

Disco: 50 / 60 pasaron 10 cil * 1 ms = 10 ms

T = 10 ms C: 115, 99

Disco: 60 / 99 pasaron 390 cil * 1ms = 39 ms + 10 ms = 49 ms

T = 49 ms C: 115, 100, 65

Disco: 99 / 100 pasaron 1 cil * 1 ms = 1 ms + 49 ms = 50 ms

T = 50 ms C: 115, 65

Disco: 100 / 115 pasaron 15 cil * 1 ms = 15 ms + 50 ms = 65 ms

T = 65 ms C: 65, 83, 125

Disco 115 / 125 pasaron 10 ms = 75 ms

T = 75 ms C: 65, 83, 25

Disco: 125 / 83 / 65 / 25 pasaron (42 cil + 18cil + 40Cil) * 1 ms = 100 ms + 75 ms = 175 ms

Orden total:

Disco: 50 / 60 / 99 / 100 / 115 / 125 / 83 / 65 / 25 .

Tiempo total de Búsqueda: 10 ms + 39 ms + 1 ms + 15 ms + 10 ms + 100 ms = 175ms

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

Otra diferencia es que N-STEP-SCAN no sufre inanición y el SSTF al llegar pedido más cercano podría sufrir inanición.

Ejercicio 2

a) El tamaño mínimo posible que un archivo podría tener sería 0 bytes, o sea que no tuviera bloques asignados. Por lo tanto, la cantidad máxima de archivos está limitada por el tamaño máximo del archivo de tipo directorio, el raíz. Se podrán tener tantos archivos como entradas de directorio puedan caber dentro de dicho archivo.

(Nota: en ext2 el verdadero límite de archivos está dado por la tabla de inodos, pero en este ejercicio no se encuentra como dato, así que puede ser ignorada)

Por lo tanto, hay que calcular el tamaño máximo de un archivo en este sistema
Tamaño máximo teórico del archivo:

$$15 \text{ p.d.} \times 4\text{Kb} + 1 \text{ p.i} \times \text{punterosEnBloque} \times 4\text{Kb} + 2 \text{ p.i.d} \times \text{punterosEnBloque}^2$$

Se calculan los punteros por bloque = $4096 \text{ bytes} / 4 \text{ bytes} = 1024$.

Repetimos la formula:

$$15 \text{ p.d.} \times 4\text{Kb} + 1 \text{ p.i} \times 1024 \times 4\text{Kb} + 2 \text{ p.i.d} \times 1024^2 \times 4\text{Kb} = 8.594.190.336 \text{ bytes} \approx 8\text{Gb}$$

Cantidad total de entradas de directorio = $8594178048 / 64 \text{ bytes} = 134.284.224$ de archivos vacíos.

b) En FAT, es similar a calcular el tamaño máximo del fs: $2^{28} \times 2^{13} \text{ bytes} = 2^{41} \text{ bytes}$

Cantidad total de entradas de directorio = $2^{41} / 32 \text{ bytes} = 68.719.476.736$ de archivos vacíos.

c) Se podría tener $(134.284.224 - 1)$ hard links, dado que como no son archivos nuevos, no necesitan tener bloques asignados.

d) Tamaño máximo teórico de la tabla FAT:

$$2^{28} \text{ entradas} \times 2^2 \text{ bytes (32 bits por entrada)} = 2^{30} \text{ bytes} = 1 \text{ Gb.}$$

Ejercicio 3

Proceso 1			Proceso 2			Proceso 3			Proceso 4		
	Frame	Bits		Frame	Bits		Frame	Bits		Frame	Bits
0	5	UMP	0	-	-	0	3	MP	0	1	-
1	2	UP	1	7	P	1	1	-	1	4	-
2	1	-	2	4	UP	2	6	MP	2	1	-
3	7	-	3	1	-	3	1	UP	3	0	P

Frame	Inicial	P2 - 0	P1 - 2	P4 - 10	<p>- Son 8 frames de 8 KB cada uno</p> <p>- Las páginas también son de 8 KB</p> <p>- La referencia anterior fue del proceso 1, página 0, dado que es la única modificada (una asignación genera una modificación) que tiene el bit de uso encendido.</p>
0	P4 - 3	→ P4 - 3	P1 - 2 U	P1 - 2 U	
1	P3 - 3 U	P3 - 3 U	→ P3 - 3 U	P3 - 3	
2	P1 - 1 U	P1 - 1 U	P1 - 1 U	P1 - 1	
3	P3 - 0 M	P3 - 0 M	P3 - 0 M	P4 - 10 U	
4	P2 - 2 U	P2 - 2 U	P2 - 2 U	→ P2 - 2 U	
5	→ P1 - 0 UM	P1 - 0 UM	P1 - 0 UM	P1 - 0 UM	
6	P3 - 2 M	P3 - 2 M	P3 - 2 M	P3 - 2 M	
7	P2 - 1	P2 - 0 U	P2 - 0 U	P2 - 0 U	