

Sistemas Operativos

2° Parcial 1C2023 – 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

	Segmentacion pura	Segmentación paginada
Estruc. admin	Una tabla de segmentos x proceso	- Una tabla de segmentos x proceso - Una tabla de páginas por segmento del proceso
Accesos por lectura	1 - Acceso a tabla de segmentos 2 - Acceso al dato en ram	1 - Acceso a tabla de segmentos 2- Acceso a tabla de páginas 3 - Acceso al dato en ram
Fragmentacion	Externa	Interna

2

Utilizar paginación jerárquica resuelve el problema de paginación tradicional de tener tablas de páginas muy grandes cargadas en memoria obligatoriamente todo el tiempo. La paginación jerárquica divide la tabla de páginas en niveles y permite cargar solamente el primer nivel en memoria e ir cargando las tablas de segundo nivel solo a medida que se referencian las mismas.

La desventaja más grande que tiene es que, al agregar más niveles, es necesario acceder a memoria más veces para poder realizar una traducción ($N - 1$ más veces que en paginación tradicional, siendo N la cantidad de niveles).

Utilizar una TLB permite mitigar este problema. Cada vez que hay un TLB hit, la traducción se realiza sin acceder a las tablas de páginas, minimizando mucho el impacto de la desventaja mencionada.

3- open()

- Verifica si los permisos el archivo permiten al usuario la apertura para lectura

- Crea o actualiza la entrada de dicho archivo en la tabla global de archivos abiertos del sistema
- Crea una entrada en la tabla de archivos abiertos del proceso, apuntando a la entrada de la tabla global. Inicializa el puntero de acceso al archivo en el byte 0 (cero)

close()

- Actualiza o elimina la entrada de dicho archivo en la tabla global de archivos abiertos
- Persiste en disco los remanentes del buffer de escritura, si existieran
- Elimina la entrada en la tabla de archivos abiertos del proceso

4

Si se tiene un archivo regular y sobre el mismo se crean un hardlink y un softlink y luego se remueve el mismo, solo se estaría removiendo la entrada de directorio y no el archivo original. Por lo tanto, se podrá seguir accediendo normalmente a través del hardlink pero no a través del softlink, ya que la referencia contiene el path del archivo original, el cuál fue removido, mientras que el hardlink apunta al archivo original a través del número de inodo.

5

1. Falso. Pasaría a corresponder a otro proceso si el segmento fuera reubicado (por ejemplo por haber sido llevado a disco y cargado nuevamente más tarde)
2. Verdadero. Esto se debe a que los archivos son cargados a memoria en páginas del sistema de paginación, por lo tanto si el mismo es múltiplo de dicho tamaño resulta más eficiente en términos de operaciones de lectura y escritura

Práctica

1)

Primero se determinan el tamaño de bloque, de puntero, y cantidad de punteros por bloque:

- Si un archivo ocupa como mínimo 2048 bytes en disco, entonces los bloques son de ese tamaño (por ser la unidad mínima de asignación)
- Si 128 bytes ocupa un inodo, y 48 bytes son para atributos, entonces $128 - 68 = 60$ bytes para punteros en el inodo. Dado que tenemos 15 punteros, entonces $60/15 = 4$ bytes para cada puntero.
- Si los bloques son de 2048 bytes y los punteros de 4 bytes, entonces $2048 / 4 = 512$ punteros por bloque

Punto a)

Se dispone de un buffer de 10240 bytes / 2048 bytes = 5 bloques.

Byte solicitado	Nro bloque en archivo	Nivel indireccion	Bloques leídos de disco	Total bloques leídos de disco
-----------------	-----------------------	-------------------	-------------------------	-------------------------------

1073153	524	doble	- 1 1er nivel - 1 2do nivel - 1 bloque dato	3
537944061	262667	doble	- <i>En buffer: 1 1er nivel</i> - 1 2do nivel - 1 dato	2
1075198	524	doble	- <i>En buffer: 1er nivel</i> - <i>En buffer: 2do nivel</i> - <i>En buffer: 1 dato</i>	0

Punto b)

Dado que el bloque a acceder es el primero de la indirección simple, primero es necesario leer el bloque de punteros para saber la ubicación del bloque de datos y poder escribir en él. No es necesario leer primero el bloque de datos, dado que se sobrescribe en su totalidad.

Byte solicitado	Nro bloque en archivo	Nivel direccion	Bloques accedidos
24576	12	simple	- 1 ind simple (leído) - 1 bloque dato (escrito)

Total: 2 bloques accedidos, 1 del disco (lectura) y otro hacia el disco (escritura)

2)

Dirección lógica AC95h => Página 10, ubicada en frame 41

Dirección física 28FC2h => Frame 40 (alberga página 8)

Página 5 en frame restante: 42

	== Estado inicial ==	Página 10	Página 6
Frame: 40	8 (u=1, m=1)	8 (u=1, m=1)	8(u=0, m=1)
Frame: 41	10 (u=0, m=0) <-	10 (u=1, m=1) <-	6(u=1, m=0)
Frame: 42	5(u=1, m=1)	5(u=1, m=1)	5(u=0, m=1) <-

Se producen 2 accesos a disco en la referencia a la página 6: 1 por page fault y otra por escribir la página 10 en disco.

3)

a)

Para poder direccionar todo el disco rígido instalado, el File System necesitará:

Cant. entradas * tamaño de bloque = Disco físico

Cant. entradas * $2^{11} = 2^{37}$

Cant. entradas = 2^{26} .

Para calcular el tamaño de la tabla FAT, debemos multiplicar la cantidad de entradas creadas por el tamaño de la entrada:

Tamaño FAT = $2^{26} * 2^2 \text{ bytes} = 2^{28} \text{ bytes} = 256\text{MiB}$.

Con marcos de 4KiB, se necesitarán:

$2^{28} \text{ bytes} / 2^{12} \text{ bytes} = 2^{16}$ marcos para poder cargar toda la tabla FAT.

Si las direcciones son de 32 bits, y se utilizan 12 bits para el offset, sabemos que nos quedan 20 bits para el número de segmento y página. Si se pueden tener 8 segmentos, sabemos que necesitamos 3 bits para el número de segmento y los otros 17 bits restantes son para la tabla de páginas. En las notas se aclara que las tablas se crean utilizando el máximo tamaño posible, por lo cual esa tabla de páginas tendrá 2^{17} entradas.

Si se necesitan 2^{16} marcos para cargar toda la tabla FAT y los mismos están todos presentes en memoria, **entonces el 50% de las entradas tienen el bit de presencia en 1.**

b) El límite de dicho segmento será del tamaño de la FAT, que es la única información cargada en el mismo, por lo cual será de 256MiB.