ASO - final - 5-8-2023

1) Calcule la fragmentación interna en una administración de memoria paginada simple, de un proceso cuyo tamaño es de 38.310 Bytes donde cada página es de 0,5 Kbytes cada una.

1. Calcule la fragmentación interna del proceso
2. Calcule la cantidad de frames que necesita el proceso
3. Calcule la cantidad de bits de la dirección lógica

a) La fragmentación interna la encontraremos en la última página de un proceso dado. Dado que su tamaño es de 38.310 bytes que deben “encajar” en páginas de 512 bytes, resulta que la última página contendrá 422 bytes de ese proceso, dejándonos así con una “sobrante libre” (fragmentación interna) de 90 bytes.

b) Dado que el proceso de 38.310 bytes está dividido en páginas de 512 bytes, para alojarlo serán necesarios 75 frames (el frame tiene el mismo tamaño que la página).

c) Teniendo en cuenta la respuestas anteriores, para direccionar 75 frames son necesarios 7 bits, y para direccionar 512 bytes de desplazamiento son necesarios 9 bits, lo que nos da 16 bits de dirección lógica.

2) En una administración de memoria paginada bajo demanda la secuencia o traza de páginas accedidas es la siguiente:

0 1 2 7 4 8 9 2 3

La cantidad de fallos de página es de 4, en todos los casos se encontró un frame libre

el tiempo de acceso a memoria es 100 nanosegundos.

el tiempo de resolución del fallo de página es de 1000 nanosegundos.

Calcule el tiempo de acceso a los datos y a las instrucciones del proceso.

(Aclaro "No es el tiempo de ejecución")

Si hacemos unos cálculos preliminares, obtendríamos esto:

* Acceso a datos (individual) = 100 nanosegundos por acceso
* Resolución de fallo de página (individual) = 1000 nanosegundos por fallo
* Cantidad de fallos de página = 4
* Tiempo total de acceso a los datos = 900 nanosegundos
* Tiempo total de resolución de fallos de página = 4000 nanosegundos
* Tiempo total de acceso a los datos y a las instrucciones del proceso = 4900 nanosegundos

Sin embargo, en el escenario dado, se menciona que "en todos los casos se encontró un frame libre", lo que significa que cuando ocurre un fallo de página, el sistema operativo no necesita buscar la página en el disco y copiarla a memoria (lo que llevaría tiempo adicional). En cambio, simplemente utiliza un frame libre disponible para cargar la página necesaria.

Entonces, dado que se asume que todos los fallos de página se resuelven sin acceder al disco, el tiempo de resolución del fallo de página no lo tengo en consideración y, por tanto, no lo incluyo en el cálculo del tiempo de acceso a los datos e instrucciones. Es decir, solo tengo en cuenta el tiempo de acceso a memoria (100 nanosegundos por acceso) para calcular el tiempo total de acceso a los datos e instrucciones.

Por tanto, por lo expuesto, mi respuesta a “calcule el tiempo de acceso a los datos y a las instrucciones del proceso” es de 900 nanosegundos.

3) En un sistema que administra el espacio en disco asignado (ocupado por los archivos) con asignación indexada o indizada, donde el disco posee una capacidad de almacenamiento de 50 MBytes y donde el bloque o registro físico o sector almacena 512 Bytes, y dado un archivo de 789 KBytes.

a) Calcule la cantidad de bloques o registros físicos de Datos del Archivo.

b) Calcule la cantidad de bloques de índice del archivo.

a) Tamaño de archivo = 789 kilobytes = 807.936 bytes.

Tamaño de un bloque físico = 512 bytes

Cantidad de bloques físicos para almacenar el archivo = 1.578 bloques

b) En el libro de Silberschatz se lee que "El mecanismo de asignación indexada agrupa todos los punteros en una única ubicación: el bloque de índice. Cada archivo tiene su propio bloque de índice, que es una matriz de direcciones de bloques de disco. La entrada í-ésima del bloque de índice apunta al bloque f-ésimo del archivo. El directorio contiene la dirección del bloque de índice. Para localizar y leer el bloque í-ésimo, utilizamos el puntero contenido en la entrada i-ésima del bloque de índice". Entonces, por tanto, la cantidad de bloques de índice del archivo es 1 (uno). Es decir, se necesita 1 (un) bloque para administrar el espacio en disco asignado bajo un sistema de asignación indexada.