

Sistemas Operativos I

Final 15/02/2011

1. Administración de memoria. Explique si las siguientes afirmaciones son verdaderas o falsas y justifique su respuesta:
 - a. La memoria virtual no necesita ningún soporte por parte del hardware para ser implementada eficientemente.
 - b. Copy on write sirve para acceder con facilidad a archivos de gran tamaño.
 - c. Los mecanismos de reubicación solo son necesarios cuando se utiliza memoria virtual.
2. Generalmente los sistemas operativos no hacen nada para evitar ni prevenir la aparición de deadlocks. ¿Se puede hacer algo para que las aplicaciones multi-hilo que uno escribe no posean deadlocks (sin modificar el núcleo del sistema operativo, bibliotecas de hilos ni el lenguaje de programación)?
3. Sistema de archivos:
 - a. ¿Qué es y para qué sirve el formateo? ¿Puede evitarse?
 - b. Bajo las mismas condiciones ¿Los sistemas de archivos basados en i-nodos poseen peor o mejor performance que los basados en FAT para accesos aleatorios? Justifique.
 - c. Explique el funcionamiento de las herramientas para detectar y corregir inconsistencias en archivos tales como scandisk (Windows) o sfck (Unix).

RESOLUCIÓN:

1.
 - a. **Falso.** La técnica de memoria virtual se implementa mediante el mecanismo de paginación por demanda o segmentación por demanda, los cuales necesitan soporte de hardware para implementarse eficientemente y correctamente.

El componente de hardware principal es el controlador MMU (unidad de gestión de memoria). Este es el encargado de mapear las direcciones lógicas o virtuales, que manejan las aplicaciones de usuario y que entiende el CPU, a direcciones físicas de la memoria principal. El MMU también hace uso de registros especiales para el acceso a las tablas de páginas de los procesos. Estos registros se llaman PTBR (registro base de tabla de páginas) y PRLR (registro longitud de tabla de páginas).

El principal objetivo del mapeo de direcciones lógicas a físicas por el MMU es el de proteger a los procesos del acceso incorrecto a los espacios de memoria de otros procesos. El MMU también mejora la eficiencia de los accesos haciendo uso de la memoria cache y TLB (buffer de traducción de vista lateral), que es una cache especial que contiene entradas de las tablas de páginas en memoria.
 - b. **Falso.** Copy on write (COW) es una técnica de optimización, que consiste en asociar a los “usuarios” los mismos punteros a iguales recursos que solicitan. A medida que estos van realizando modificaciones a los recursos, el sistema realiza copias privadas de los recursos para prevenir los cambios a los demás “usuarios”. Esto se realiza transparentemente.

COW se utiliza en:

- Memoria virtual: cuando un proceso crea una copia de si mismo (fork en UNIX), las páginas del proceso son marcadas como Copy on write, y tanto el proceso padre e hijo comparten las mismas paginas. A medida que el proceso hijo bifurca su ejecución y realiza modificaciones a las paginas compartidas, se lanzan interrupciones (trap) que son capturadas por el SO quien realiza las copias de las paginas a modificar.
 - Compartir librerías de enlace dinámico (DLL), etc.
- c. **Falso.** Los mecanismos de reubicación son necesarios en todos los sistemas operativos multiprogramados, ya sea que su mecanismo de asignación de memoria sea contiguo, por paginación o por segmentación.
- Si no se permitiera la reubicación de procesos, solo un proceso (si la vinculación de direcciones es en tiempo de compilación) o unos pocos (si es en tiempo de carga) podrían cargarse a la vez en la sección de memoria del usuario. Así se deberían intercambiar procesos enteros entre la memoria y el almacenamiento secundario, entorpeciendo enormemente la performance de los sistemas multiprogramados.
2. Se pueden aplicar los métodos para prevenir y evitar los bloqueos mutuos por parte del programador.
- La prevención de bloqueo mutuos se realiza asegurando que por lo menos una de las condiciones necesarias no se cumpla.
 - La evitación de bloqueo mutuos por su parte, requiere que el programador implemente un mecanismo auxiliar que utilice la información de los hilos principales de la aplicación para decidir que hilo debe esperar y cual no, manteniendo el estado seguro del sistema.
- 3.
- a. El formateo es un conjunto de operaciones que permiten restablecer un disco duro, una partición del mismo o cualquier otro dispositivo de almacenamiento de datos a su estado original. Existen distintos tipos de formateos:
- Formateo físico o de bajo nivel: este tipo de formateo llena el disco con una estructura de datos especial para cada sector (o bloque). Esta estructura de datos consiste en una cabecera, un area de datos asociada al sector (512 bytes generalmente, pero se pueden especificar otros tamaños de bloques), y un tráiler. La cabeza y el tráiler contienen información usada por el controlador de disco por cuestiones de eficiencia y confiabilidad, como el numero de sector y código corrector de errores (ECC). El ECC es utilizado por el controlador de disco tanto para detectar y corregir errores de los bits del sector.
 - Particionado de disco: se considera un paso intermedio entre el formateo físico y lógico de discos. Consiste en particionar el disco en uno o más grupos de cilindros. Asi, el SO trata cada partición como un disco separada que puede operar de distintos modos o darles distintos formateos lógicos.
 - Formateo de alto nivel o lógico: consiste en “hacer el sistema de archivos” sobre una partición. Este formateo es realizado por el SO y consiste en almacenar la estructura de datos

del sistema de archivos en el disco, como también verificar posibles errores físicos que pueda tener.

El formateo físico no puede evitarse. Los discos duros vienen formateados de fábrica. Los discos flexibles necesitan ser formateados a bajo nivel por el SO cuando se dañan por interferencias electromagnéticas.

El formateo lógico puede evitarse. Algunos SOs dan a programas especiales la habilidad de usar una partición del disco como un gran arreglo secuencial de bloques lógicos sin ninguna estructura de datos del sistema de archivos. Por ejemplo, para algunos sistemas de bases de datos, ya que les permite el control de la ubicación exacta del disco donde cada registro de la base de datos esta almacenado, y particiones de intercambio (SWAP), para la administración de memoria virtual.

- b. Los sistemas de archivos basados en i-nodos tienen mejor performance que los basados en FAT para acceso aleatorio (o directo). Esto se debe a que el primero se basa en los métodos de asignación indexados, que se caracterizan por un buen acceso aleatorio a los datos, mientras que FAT se basa en los métodos de asignación enlazada, que posee un muy buen acceso secuencial a los datos pero un pobre acceso directo.

Para entender esto, veamos el caso de un acceso directo al final de un archivo de tamaño grande (varios Gigabytes). En el caso de i-nodes, como máximo, se pueden requerir 4 accesos a disco, uno al bloque i-node, 2 a bloques indirectos de índices, y uno más al bloque de datos. En el caso de FAT, resulta en un número considerable de búsquedas de la cabeza de disco, a menos que la FAT se coloque en cache.

Estos programas se ejecutan en el momento de reinicio del sistema para chequear inconsistencias del disco generadas, posiblemente, por la caída del sistema. En una caída accidental del sistema, la tabla de archivos abiertos y otras estructuras de datos en memoria principal, asociados a los metadatos de los archivos y la estructura de directorio, se pierden, y con este, todos los posibles cambios realizados a los directorios de archivos abiertos.

El chequeo de consistencia compara el dato en la estructura de directorio con el dato en los bloques de disco, y trata de arreglar, forzosamente, cualquier inconsistencia que encuentre.

Los nuevos sistemas de archivos implementan nuevas técnicas y algoritmos por lo que son menos propensos a estos problemas de inconsistencia. Por ejemplo, NTFS, ext3 (Linux) y algunas versiones modernas de UFS son orientados a transacciones, asegurándose de escribir primero en un archivo de registros en disco las modificaciones asociadas a los metadatos de archivos y estructura de directorio. Estos sistemas de archivos no requieren así un chequeo de disco, como los antiguos sistemas FAT, incluso después de un apagado incorrecto del sistema. Esto es debido a que los datos son escritos al disco de tal manera que el sistema de ficheros siempre esta consistente, o se puede recuperar fácilmente su consistencia por el archivo de registros. Sólo se realizará un "fsck" o "scandisk" en el caso de fallos hardware raramente dados (por ejemplo fallos físicos del disco duro).