LICENCIATURA EN SISTEMAS DE LA INFORMACION

TP 5
FILESYSTEMS

UADER FCyT

SISTEMAS OPERATIVOS

ALUMNOS

ERRANDONEA GONZALO
ROMERO GONZALO
ALALUF TOMAS

PROFESORES

OSVALDO AGUIAR
ULISES RAPALLINI

02/09/2022

Trabajo de investigación: Directorios – Filesystems

Bibliografías recomendadas:

- Sistemas Operativos. Un enfoque en espiral – Ramez Elmasri y otros.
- Sistemas operativos. Aspectos internos y principios de diseño – William Stallings

1. **¿Qué funciones se pueden llevar a cabo sobre un archivo? ¿Qué conjunto de atributos se asocian a un archivo?**

   Cualquier sistema de ficheros proporciona no sólo una manera de almacenar los datos organizados como ficheros, sino una colección de funciones que se pueden llevar a cabo sobre ficheros. Algunas operaciones típicas son las siguientes:

   • **Crear.** Se define un nuevo fichero y se posiciona dentro de la estructura de ficheros.

   • **Borrar.** Se elimina un fichero de la estructura de ficheros y se destruye.

   • **Abrir.** Un fichero existente se declara «abierto» por un proceso, permitiendo al proceso realizar funciones sobre dicho fichero.

   • **Cerrar.** Un determinado proceso cierra un fichero, de forma que no puede volver a realizar

   • **Leer.** Un proceso lee de un fichero todos los datos o una porción de ellos.

   • **Escribir.** Un proceso actualiza un fichero, bien añadiendo nuevos datos que expanden el tamaño del fichero, bien cambiando los valores de elementos de datos existentes en el fichero.

   Típicamente, un sistema de ficheros mantiene un conjunto de atributos asociados al fichero. Estos incluyen el propietario, tiempo de creación, tiempo de última modificación, privilegios de acceso, etc.

2. **¿Qué operaciones típicas deberán soportar las aplicaciones que deseen utilizar archivos?**

   Los usuarios y las aplicaciones desean utilizar ficheros. Las operaciones típicas que deben soportarse incluyen:

• Obtener_Todos. Obtener todos los registros de un fichero. Esta operación se requerirá por aplicaciones que deban procesar toda la información de un fichero de una vez. Por ejemplo, una aplicación que produzca un resumen de la información existente en un fichero necesitaría obtener todos sus registros. Esta operación se asocia frecuentemente con el término procesamiento secuencial, porque todos los registros se acceden en secuencia.
• Obtener_Uno. Esta operación solicita un único registro. Las aplicaciones interactivas y orientadas a transacciones necesitan esta operación.
• Obtener_Siguiente. Esta operación solicita el «siguiente» registro, en alguna secuencia lógica con respecto al registro más recientemente leído. Algunas aplicaciones interactivas, tales como el rellenado de formularios, podrían requerir este tipo de operaciones. Un programa que realiza una búsqueda podría también utilizar este tipo de operación.
• Obtener_Anterior. Similar a la operación Obtener_Siguiente, pero en este caso el registro al
que se accede es el anterior al más recientemente leído.
• Insertar_Uno. Insertar un nuevo registro en el fichero. Puede ser necesario que el nuevo registro encaje en una posición específica, para preservar una secuencia del fichero.
• Borrar_Uno. Borrar un registro existente. Ciertos enlaces u otras estructuras de datos podrían
tener que actualizarse para preservar la secuencia del fichero.
• Actualizar_Uno. Obtener un registro, actualizar uno o más de sus campos, y reescribir el registro actualizado en el fichero. De nuevo, puede ser necesario preservar la secuenciación con
esta operación. Si la longitud del registro ha cambiado, la operación de actualización es generalmente más difícil que si se preserva la longitud.
• Obtener_Varios. Obtener varios registros. Por ejemplo, una aplicación o usuario podría desear obtener todos los registros que satisfacen un cierto conjunto de criterios.

3. **¿En qué consiste un sistema de gestión de archivos y cuáles son los objetivos?**
Un sistema de gestión de ficheros es aquel conjunto de software de sistema que proporciona servicios a los usuarios y aplicaciones en el uso de ficheros. Típicamente, la única forma en la que un usuario o aplicación puede acceder a los ficheros es a través del sistema de gestión de ficheros. Esto elimina la necesidad de que el usuario o programador desarrolle software de propósito especial para cada aplicación. Además, proporciona al sistema una forma consistente y bien definida de controlar su recurso más importante.
Sus objetivos son:
• **Satisfacer las necesidades** de gestión de datos y requisitos del usuario, lo que incluye el

almacenamiento de datos y la capacidad de llevar a cabo las operaciones anteriormente mencionadas.
• **Garantizar**, hasta donde sea posible, que los datos del fichero son válidos.
• **Optimizar el rendimiento**, desde el punto de vista del sistema en términos de productividad y desde el punto de vista del usuario en términos de tiempo de respuesta.
• **Proporcionar soporte de E/S** a una variedad de tipos de dispositivos de almacenamiento.
• **Minimizar o eliminar** la potencial pérdida de datos.
• **Proporcionar un conjunto estándar de rutin**as de interfaces de E/S a los procesos.
• **Proporcionar soporte de E/S a múltiples usuarios**, en el caso de sistemas multiusuarios.

4. **¿Cuáles son los requisitos mínimos para un sistema interactivo de propósito general?**

   Para un sistema interactivo, de propósito general, las siguientes características constituyen un conjunto mínimo de requisitos:

   1. Cada usuario debería poder crear, borrar, leer, escribir y modificar ficheros.

   2. Cada usuario tendría acceso controlado a los ficheros de otros usuarios.

   3. Cada usuario podría controlar qué tipos de accesos se permiten a los ficheros de los usuarios.

   4. Cada usuario debe poder reestructurar los ficheros de los usuarios en una forma apropiada al problema.

   5. Cada usuario debe ser capaz de mover datos entre ficheros.

   6. Cada usuario debe poder copiar y recuperar los ficheros de usuario en caso de daño.

   7. Cada usuario debe poder acceder a los ficheros utilizando nombres simbólicos.

5. **¿Qué criterios se tienen en cuenta para escoger una organización de archivos?**
   Para escoger una organización de ficheros, son importantes varios criterios:
   • **Tiempo de acceso corto.**
   • **Facilidad de actualización.**
   • **Economía de almacenamiento.**
   • **Mantenimiento sencillo.**
   • **Fiabilidad.**
   La prioridad relativa de estos criterios dependerá de las aplicaciones que utilizarán el fichero. Por ejemplo, si un fichero se va a procesar sólo en lotes, con acceso a todos los registros cada vez, entonces el acceso rápido a un único registro no es un requisito. Un fichero almacenado en CD-ROM nunca se actualizará, y por tanto la facilidad de actualización no es un aspecto a tener en cuenta en este caso.

6. **¿Qué es un directorio? ¿Qué son los metadatos de un archivo?**

   Un **directorio** es un objeto que relaciona de forma unívoca el nombre de un archivo y el descriptor interno del mismo usado por el sistema operativo. Los directorios sirven para organizar y proporcionar información acerca de la estructuración de los archivos en los sistemas de archivos.

Los **metadatos** de un archivo son toda la información auxiliar que es necesario mantener para ofrecer la visión lógica de un archivo. Esta información incluye entre otros, los bloques que ocupan el archivo en disco.

7. **¿Qué operaciones sobre los directorios atiende el SO?**
   Esquema de dos niveles. Hay un directorio por cada usuario y un directorio maestro. El directorio maestro tiene una entrada por cada directorio usuario, proporcionando información sobre dirección y control de acceso. Cada directorio de usuario es una lista simple de los ficheros de dicho usuario. Esto implica que los nombres deben ser únicos sólo dentro de la colección de los ficheros de un único usuario y que el sistema de ficheros puede fácilmente asegurar las restricciones de acceso de los directorios.
   Estructura jerárquica en forma de árbol. Como en la técnica anterior, hay un directorio maestro, que tiene bajo dicho directorio varios directorios de usuario. Cada uno de estos directorios de usuario, a su vez, podría tener subdirectorios y ficheros como entradas. Esto se cumple para todos los niveles: es decir, en cada nivel, un directorio podría estar formado por subdirectorios y/o ficheros.

8. **Enumere los métodos de acceso básicos que proveen los Sistemas Operativos. ¿Qué es el "acceso en bruto"? ¿Quiénes podrían necesitarlo?**

Los métodos de acceso básico que proveen los Sistemas Operativos son:

- **Acceso secuencial**
- **Acceso aleatorio**
- **Fichero indexado**
- **Acceso directo o hash**

En el **acceso en bruto**, el S.O no suministra ninguna estructura de archivos, sino que reserva un área del disco donde las aplicaciones pueden proporcionar su propia estructura. Algunos ejemplos de aplicaciones en las que estos accesos en bruto son útiles incluyen el almacén de paginación del Sistema Operativo— en sí y sistemas de bases de datos

9. **¿Qué es un bloque? ¿por qué surge? ¿Qué inconvenientes tiene esta estructura?**

   Un **bloque** se define como una agrupación lógica de sectores de disco y es la unidad mínima que usa el sistema de archivos. Se usa para optimizar la eficiencia de la E/S de los dispositivos secundarios de almacenamiento.
   El problema que se presenta es que, los bloques que son grandes, son propensos a tener fragmentación interna

10. Explique las diferentes maneras de rastrear el espacio libre en un dispositivo de disco.
Tablas de bits. Este método utiliza un vector que está formado por un bit por cada bloque en el disco. Cada entrada 0 corresponde a un bloque libre y cada 1 corresponde a un bloque en uso.
Porciones libres encadenadas. Las porciones libres se pueden encadenar utilizando un puntero y valor de longitud en cada porción libre. Este método tiene una sobrecarga de espacio

insignificante, porque no se necesita una tabla de asignación de disco, sino simplemente un puntero al comienzo de la cadena y la longitud de la primera porción.

Indexación. La técnica de indexación trata el espacio libre como un fichero y utiliza una tabla de índices tal y como se describió en la asignación de ficheros. Por motivos de eficiencia, el índice se debería utilizar en base a porciones de tamaño variable en lugar de bloques. Por tanto, hay una entrada en la tabla por cada porción libre en el disco.

Lista de bloques libres. En este método, a cada bloque se le asigna un número secuencialmente y la lista de los números de todos los bloques libres se mantiene en una porción reservada del disco. Dependiendo del tamaño del disco, se necesitarán 24 o 32 bits para almacenar un único número de bloque, de tal forma que el tamaño de la lista de bloques libres es 24 o 32 veces el tamaño de la correspondiente tabla de bits y por tanto debe almacenarse en disco y no en memoria principal.

11. ¿En qué consiste el método de asignación contigua? ¿Qué problemas presenta?

En el método de **asignación contigua**, se asigna un único conjunto contiguo de bloques en tiempo de creación de los ficheros. Por tanto, hay una estrategia de pre-asignación que utiliza porciones de tamaño variable. La tabla de asignación de ficheros necesita sólo una entrada para cada fichero, mostrando el bloque inicial y la longitud del fichero.
La asignación contigua presenta algunos problemas. Existirá fragmentación externa, haciendo difícil encontrar bloques contiguos de espacio de suficiente longitud. De vez en cuando, será necesario llevar a cabo un algoritmo de compactación para liberar espacio adicional en el disco. Además, con preasignación, es necesario declarar el tamaño del fichero en el tiempo de creación, con los problemas mencionados anteriormente.

12. ¿En qué consiste la desfragmentación? ¿Qué tipos de fragmentación soluciona?

La idea básica de la **desfragmentación** es mover algunos archivos hacia huecos donde quepan dejando huecos más grandes para los archivos que queremos asignar. Esto soluciona la fragmentación externa.

13. ¿Qué es la asignación vinculada? ¿qué inconvenientes tiene?

El mecanismo de **asignación vinculada** es como una estructura de linked list en memoria primaria, pero aquí los elementos vinculados siempre son del mismo tamaño: un bloque en el archivo. Cada bloque contiene la dirección del sector de inicio del siguiente bloque en el archivo.
Un inconveniente de este mecanismo es que en este vínculo se desperdicia una parte en cada bloque. Otro inconveniente es que algunas veces es difícil efectuar métodos de acceso aleatorio en tales archivos, aunque no es imposibles.

14. ¿Qué es la asignación indexada? ¿cómo se implementa?

La **asignación indexada** resuelve muchos de los problemas de la asignación contigua y encadenada. En este caso, la tabla de asignación de ficheros contiene un índice separado de un nivel por cada fichero; el índice tiene una entrada por cada porción asignada al fichero.

Típicamente, los índices de fichero no se almacenan físicamente como parte de la tabla de asignación de ficheros. Por el contrario, el índice de ficheros para un fichero se guarda en un bloque separado y la entrada para fichero en la tabla de asignación de ficheros apunta a dicho bloque.
La asignación puede implementarse mediante bloques de tamaño fijo o porciones de tamaño variable. La asignación por bloques elimina la fragmentación externa, mientras que la asignación por porciones de tamaño variable mejora la proximidad.