Gestion Ficheros

**Fichero**—Un fichero es una abstracción de almacenamiento secindario, sucesión de bytes cuya semántica no afecta al SSOO. Debe cumplir 3 condiciones-Almacenar info ilimitada, sobrevivir al proceso que lo creó y permitir acceso simul de varios procesos.

Disco magnético:unidad mínima lectura/escrit-Sector(típico 512 bytes)—conjunto de sectores concéntricos-pista

Unidad mínima de transferencia entre disco y ordenador—Bloque—agrupacion(conjunto de bloques)

Mapa de fichero—mapa de agrupaciones que ocupa..Puede generar fragmentación interna.

Metadatos de un fichero:Identificador—Nombre del fichero—Fechas de creación ultimo acceso y mod—tipo—mapa del fichero—propietario—proteccion—tamaño—Estos metadatos se guardan en DFF(descriptor físico de fichero) en window se llama MFT (master table file), en unix i-nodo. Un directorio es un fichero especial que guarda información de otros ficheros (encontramos la relación entre el nombre del fichero y su DFF). Los subdirectorios permiten agrupar ficheros. Los directorios actuales son estructuras en árbol que apuntan a ficheros o subdirectorios.

Tabla

Descripción generada automáticamenteCada fichero libre tuene un nombre que debe de ser único en su carpeta. En el directorio cada fichero tiene su nombre y su i-nodo. Cada directorio tiene entradas especiales que apuntan a su padre y a sí mismo, “.”es para sí mismo y “..”es para su padre. El directorio raíz es su propio padre para evitar errores de correlación.

i-nodo->guarda info sobre bloques—bloques guardan info del fichero

Soft link🡪entrada en un directorio que apunta a otro fichero

**Sistema de ficheros:**Estructura de info que permite maejar los ficheros de una partición (unidad lógica en windows).El sistema se crea al formatear la partición. En todo volumen(USB, disco,etc) sector 0 es el MBR que tiene 512 bytes, este no forma parte del sistema. Al final del MBR (últimos 64 bytes) esta la tabla de particiones. MBR solo hasta 2TB y 4 particiones—GPT casi ilimitado de particiones. En cada partición se reserva el bloque de arranque al principio y luego tenemos el superbloque que contiene info sobre todo el sistema:numero de inodos, numero de bloques bitmaps de bloques e inodos libres etc. Hay bitmap de bloques y otro de inodos que tienen tantos bits como bloques o inodos haya y un 1 o un 0 dependiendo de si existe o no el bloque o inodo en esa posición.

Inodos—Indirección ya que con una entrada de 12 bloques en cada inodo solo se podían hacer ficheros de 48kB que no es nada. Se añadió un “puntero” a la entrada que contenía el número del bloque donde se guardaban los números de bloque adicionales. Se llegan a hacer hasta indirecciones triples.

Extent—rango de bloques que se pueden asignar en ext4

FAT—file allocate table Sistema de registro de los bloques en MS-DOS. En vez de bitmap esto usa una lista enlazada de bloques. Cada entrada del vector es 0 si el bloque está libre, el numero del siguiente bloque ocupado si esta ocupado o -1 si es el fina EOF.

Sistema Virtual de Ficheros es otra capa más para que sea invisible para el programador que unifica el acceso a todas las particiones entre volúmenes.

Servidor de Ficheros: Parte del SSOO que ofrece al programador la abstracción FILE. Para abrir un fichero el sistema recorre el directorio hasta encontrar el nombre del fichero, si no existe devuelve NULL, después comprueba que el user tiene permisos, si es así el inodo se instancia en la tabla de inodos del Kernel se añade una entrada en la tabla global de ficheros y se escribe el numero de entrada en el BCP del proceso que invocó fopen()

Para crear primero comprueba que el nombre no exista ya y los permisos para crearlo, se busca un idono libre y se instancia en memoria, se devuelve el id en la tabla intermedia del procelso llamante.

El servicio de lectura requiere un descriptor de fichero abierto, el número de bytes (N) quese van a leer y la dirección en que se almacenarán. Si todo es correcto el puntero del ficherose incrementa en N.

Este servicio permite escribir M bytes, devuelve el número de bytes realmente escrito. Elservicio se tiene que ocupar de buscar agrupaciones libres (con el bitmap de agrupaciones) para poder ampliar el fichero actual, si es el caso

El SSOO se ocupa de invocar el cierre del fichero.

El montaje es el proceso de generar a partir del directorio raíz una jerarquía de árbol.

La caché de ficheros es una zona de RAM que sí gestiona el Sistema Operativo de forma directa y utiliza la misma propiedad de proximidad referencial que la caché de memoria o que la memoria virtual. Se trata de reducir al mínimo los accesos a disco físico.

Políticas de sincronización:

-Write through (inmediata): Un bloque se escribe en cuanto se modifica, equivale que no haya caché.

-Write Back: Los datos solo se escriben desde la caché a disco cuando ésta se ha llenado y hay que reemplazarlos. Si durante la ejecución hay varios cambios en un mismo bloque solo se reflejan en la caché, por lo que podrían perderse si se pierde la alimentación.

-Delayed write: : La sincronización se produce de forma periódica, lo que supone un compromiso entre los dos modos anteriores y reduce el riesgo de pérdida de información

WWrite on Close: En el cierre de un fichero siempre se produce una sincronización, con independencia de la política usada. Con write on close, la diferencia es que solo se escribe en disco al cierre, no antes

Journaling—es un diario de transiciones por si hay algún problema de alimentación no perder la información.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Una captura de pantalla de un celular con texto

Descripción generada automáticamente con confianza media