

Curso: Sistemas Operativos

5. Sistema de Archivos

(Parte I)

Sistema de Archivos

El Sistema de Archivos se ocupa de administrar el espacio de almacenamiento secundario, primordialmente en discos; así como de la transferencia de archivos.

En los inicios de los sistemas operativos, los medios de almacenamiento secundario eran en dispositivos meramente secuenciales como cintas magnéticas y de papel, tarjetas perforadas, etc. Posteriormente surgieron los discos magnéticos, los cuales han ido evolucionando tanto en materiales, tamaños y capacidades en el transcurso de 3 décadas; los discos magnéticos son dispositivos giratorios de acceso aleatorio.

Si bien, en los últimos años se ha ocupado almacenamiento de estado sólido (sin partes móviles, como memoria flash), los sistemas de archivos están generalmente desarrollados pensando en discos; a pesar de que los medios de almacenamiento de estado sólido son fundamentalmente distintos a los discos magnéticos, se presentan ante el sistema operativo como si fueran lo mismo, ya que estas unidades cuentan con una capa de traducción que emula el comportamiento de un disco duro.

Es por lo anterior que en este tema, nos enfocaremos a los discos como medio de almacenamiento secundario para la administración de archivos.

Sistema de Archivos. Conceptos básicos.

Archivo. Es un conjunto de datos al cual se le asigna un nombre. Los datos pueden ser: texto, código, video, audio, etc. Sin embargo, el sistema de archivos considera a los datos como bloques de bytes. Es la aplicación correspondiente la que permite producir, editar y acceder en general al archivo.

Las operaciones que generalmente el sistema de archivos puede realizar sobre un archivo son: abrir, cerrar, crear, destruir, copiar, renombrar, listar, ejecutar, entre otras. Muchas de estas operaciones se realizan a través de un intérprete de comandos, otras a través de llamadas al sistema.

Los datos de los archivos se pueden manipular con operaciones tales como: leer, escribir, modificar, insertar, eliminar, etc. Las cuales pueden realizarse a través de llamadas al sistema o bien de compiladores e intérpretes.

División de archivos en bloques. Como se mencionó arriba, los datos de un archivo son vistos por el sistema de archivos como bloques de bytes; pero veamos de manera más descriptiva, cuál es la jerarquía de datos en un archivo.

Sabemos que los bits (unidad básica de almacenamiento) se agrupan en *bytes* para representar un dato básico, como caracteres, pixeles, etc. La siguiente jerarquía está representada por un *campo*, el cual está formado por un grupo de bytes o caracteres. Y a su vez, un conjunto de campos define un *registro*.

Sistema de Archivos. Conceptos básicos.

División de archivos en bloques (continuación).

Por último, el conjunto de registros es el que conforma un archivo. Por ejemplo, un archivo que contiene datos de los estudiantes de alguna escuela; cada registro contiene datos de un solo estudiante. A su vez, el registro contiene campos donde en cada uno hay un dato propio de dicho estudiante, como puede ser su nombre, edad, domicilio, sexo, etc. Dichos campos o datos están formados por caracteres o números representados en bytes; que a su vez son un conjunto de bits.

Los usuarios de archivos del tipo del ejemplo planteado, ven los datos a nivel registro/campo. Sin embargo el sistema de archivos divide el archivo en *bloques de bytes*. Presentaré qué tipos de bloques maneja el sistema de archivos y su comparación con cómo los ve/maneja el usuario.

- Registro físico o bloque. Es la unidad de información que se lee realmente de un dispositivo o se graba en él.
- Registro lógico. Es un conjunto de datos/campos manejado como una unidad vista por el usuario.
- Registro sin bloques. Cada registro físico contiene un solo registro lógico.
- Registro en bloques. Cada registro físico contiene varios registros lógicos.

Los últimos dos son referidos a archivos con tamaño de registro fijo.

5 Sistema de Archivos

5.1 Introducción

Ejercicio 1

Un cierto sistema de archivos maneja un registro físico o bloque de 1024 bytes; si tiene que hacer la lectura completa de los siguientes archivos, indica por cada archivo cuántas transferencias tiene que hacer, si se tratan de registro sin bloques o en bloques y si se presentan fragmentos.

Archivo	Tamaño archivo(bytes)	Tamaño registro lógico
Arch1	12,500	250
Arch2	880	80
Arch3	24,240	1010

Respuesta

Sabemos que cuando se hace una lectura o escritura de o a un archivo, se realiza por medio de bloques, no se realiza byte a byte. Por lo que debemos ver en cuántos bloques se divide cada archivo, considerando que deben entrar registros lógicos completos. El fragmento se presenta si no se cubre todo el bloque con información del archivo; esto sucede cuando el tamaño del bloque no es múltiplo del tamaño del registro lógico. También hay que considerar que el último bloque puede tener menos registros lógicos.

Ejercicio 1

Respuesta (continuación).

Para Arch1:

El número de registros lógicos por bloque = $\text{trunc}(1024/250) = 4$; por lo que se trata de registro en bloque.

El archivo está formado por 50 registros ($12500/250$), entonces hay 12 bloques con 4 registros lógicos y 1 un bloque con 2 registros lógicos

Los primeros 12 bloques tienen un fragmento de 24 bytes y el último de 524 bytes.

Para Arch2:

Como el tamaño del archivo es menor al tamaño del bloque ($880 < 1024$), se transfiere completo en un solo bloque. Con un fragmento de 114 bytes.

El archivo está formado por 11 registros ($880/80$). Entonces se trata de registro en bloques.

Para Arch3:

El número de registros lógicos por bloque = $\text{trunc}(1024/1010) = 1$; por lo que se trata de registro sin bloque

El archivo está formado por 24 registros ($24240/1010$), todos los bloques tienen un fragmento de 14 bytes.

Sistema de Archivos. Funciones.

Funciones de los sistemas de archivos. Estas son algunas funciones que son normalmente atribuidas a los sistemas de archivos:

- Los usuarios deben poder crear, modificar y eliminar archivos.
- Los usuarios deben ser capaces de compartir sus archivos entre sí en forma cuidadosamente controlada para aprovechar y continuar el trabajo de los demás. Por lo que debe tener mecanismos de acceso controlado para lectura, escritura, ejecución y combinación de ellas.
- Los usuarios deben poder dar a sus archivos la estructura más adecuada para cada aplicación.
- Los usuarios deben ser capaces de ordenar la transferencia de información entre archivos.
- Deben ofrecer recursos de respaldo y recuperación para evitar ya sea la pérdida accidental o la destrucción mal intencionada de la información.
- Los usuarios deben poder hacer referencia a sus archivos mediante nombres simbólicos en vez de tener que usar nombres de dispositivos físicos.
- Deben ofrecer funciones de ciframiento y desciframiento en la transferencia de información para que sólo sea vista por los usuarios permitidos.
- Deben ofrecer una interfaz amable con el usuario. Proporcionar a los usuarios una vista lógica de sus datos y de las funciones que se pueden realizar con ellos.

Sistema de Archivos. Conceptos básicos.

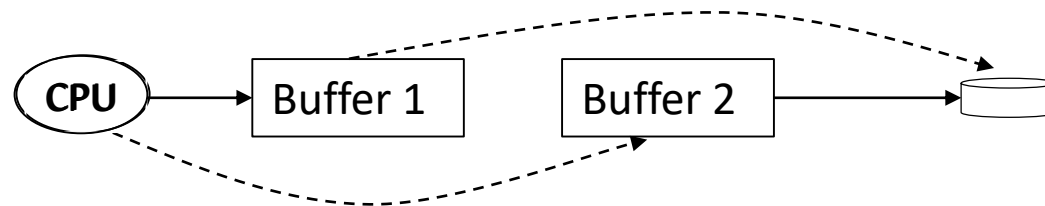
Uso de buffers.

Como la lectura y escritura de archivos se realiza por bloques, existe un elemento de almacenamiento intermedio que agiliza la transferencia de bloques. Este elemento de almacenamiento intermedio es el *buffer*; en éste se pueden almacenar varios bloques.

En el caso de operaciones de salida, el *buffer* proporciona un lugar de espera donde los bloques pueden estar hasta que el dispositivo (generalmente más lento) los graba. Esto permite que la CPU pueda trabajar en otras tareas mientras espera que el dispositivo más lento acabe de realizar el trabajo. Además el proceso sólo está en estado de bloqueo cuando el bloque ya se encuentra en el buffer y no en el dispositivo de salida.

Pueden ser varios buffers para agilizar aún más la operación, donde si un buffer está lleno o aún no se termina de vaciar al disco, se puede ocupar otro.

A esta técnica se le llama **spooling** (de las siglas de *simultaneous peripheral operations on-line*)



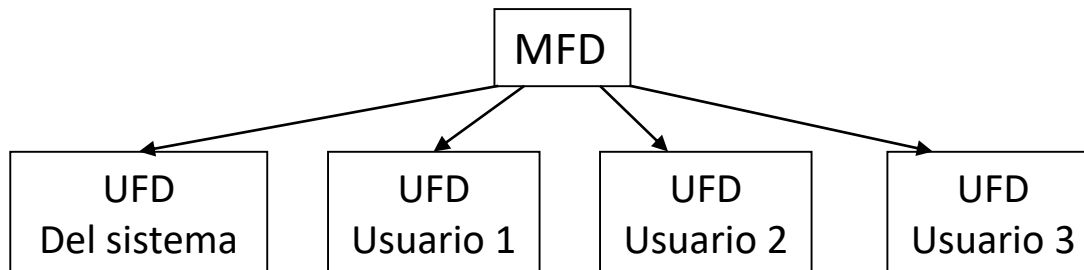
5 Sistema de Archivos

5.1 Organización de archivos

Organización de archivos.

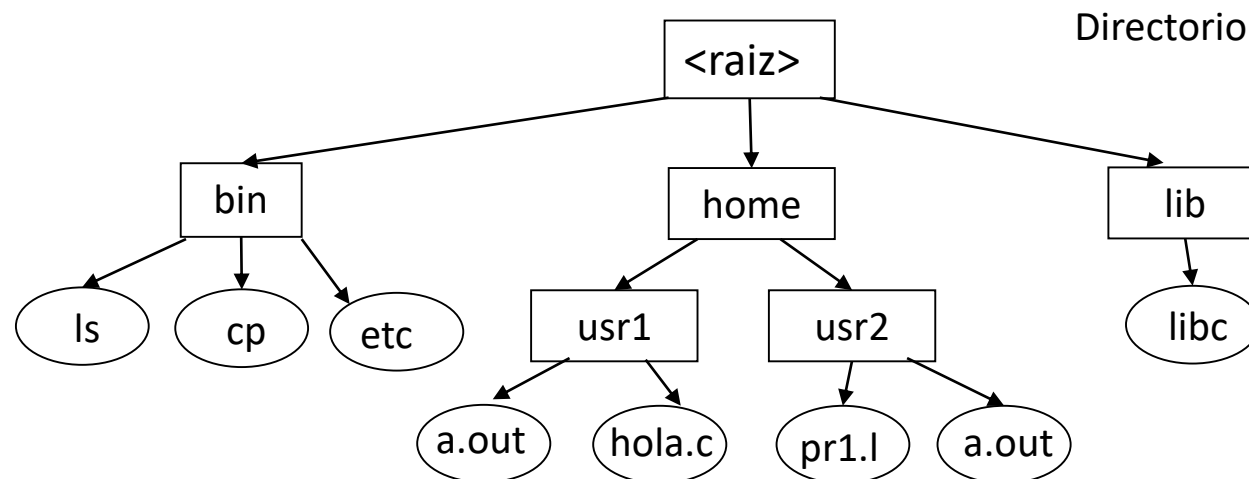
Se refiere a cómo conviven los diferentes archivos en el medio de almacenamiento.

- *Sistema de archivos plano.* Se tenía en los primeros sistemas operativos. El concepto de directorio no era empleado para clasificar a los archivos, como se hace ahora. En esta organización sólo existe un directorio cuya función es mantener una lista de los archivos que se encuentran en el dispositivo y su ubicación. También se implementó en los primeros sistemas de computadoras personales por el sistema operativo básico que tenían (por ejemplo MS-DOS y CP/M). Aún se aplica en sistemas actuales como Amazon S3 en su servicio de almacenamiento en la nube.
- *Directorios de profundidad fija.* Implementación de directorios de un solo nivel. Los archivos del sistema podían estar clasificados en directorios, ya fuera por usuario y/o por tipo de archivo (del sistema y del usuario). En esta organización existe un nodo raíz cuyo nombre es MFD (master file directory) y cada uno de los directorios derivados se les llama UFD (user file directory).



Organización de archivos.

- *Directorios estructurados en árbol.* Cuando surgen los sistemas de tiempo compartido a gran escala y en sistemas de procesamiento distribuido, tanto el número de usuarios como sus archivos aumentaron considerablemente. Esto conllevó a que en un disco hubieran decenas o cientos de miles de archivos. Por lo que, un reto del sistema de archivos, era organizarlos de forma eficiente para que cada usuario pudiera acceder a sus archivos de una forma fácil y rápida. Es así que surgen los sistemas de archivos jerárquicos sin *límite* en el número de niveles. Esto permite que cada usuario (y el administrador del sistema) estructure su información siguiendo criterios lógicos. Cuenta con un directorio raíz de donde parte toda la jerarquía de directorios.



Directorio estructurado en árbol

Nota: los directorios están representados en cuadros.

5 Sistema de Archivos

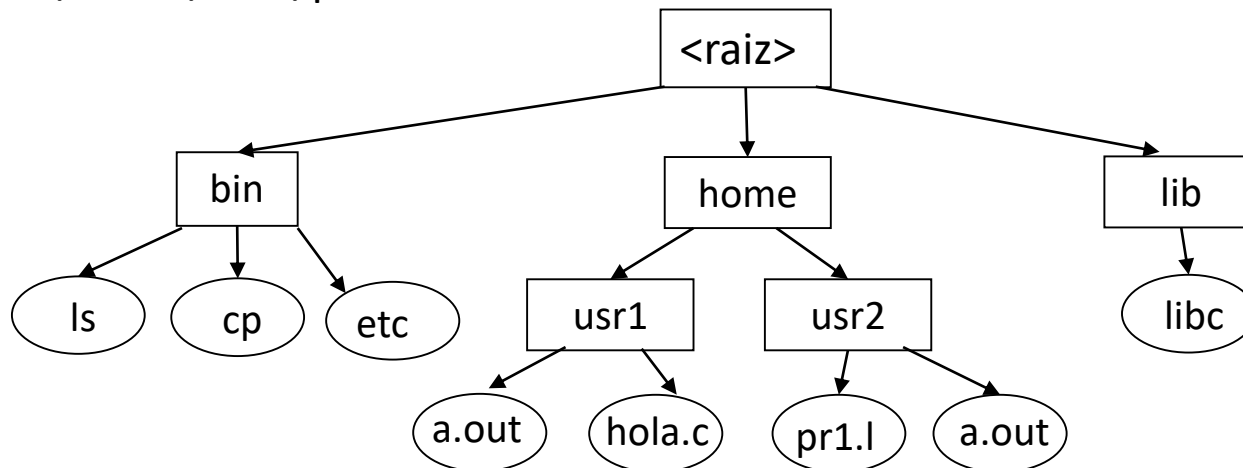
5.1 Organización de archivos

Organización de archivos.

- *Directorios estructurados en árbol (continuación).*

Del diagrama, podemos observar que un directorio puede contener a otros archivos o directorios anidados. Se pueden tener archivos con el mismo nombre, siempre y cuando no pertenezcan al mismo directorio (nótese el caso del archivo “a.out”. En realidad, el nombre completo de un archivo, en esta organización, incluye la *ruta de ubicación* donde indica su localización dentro de esta jerarquía.

Por ejemplo en Linux, la ruta o *path* absoluta inicia en la raíz indicada por ‘/’, y le siguen los nombres de los directorios en línea directa al archivo, separados por / hasta llegar al archivo. Del diagrama, el nombre completo del programa pr1.l es: /home/usr2/pr1.l



Sistema de Archivos. Componentes del sistema de archivos.

Componentes del sistema de archivos. En general, un sistema de archivos contiene:

- Métodos de acceso: Se ocupan de la forma en que se obtiene acceso a los datos almacenados en archivos.
- Administración de archivos. Ofrece mecanismo para almacenar, compartir y asegurar archivos, y hacer referencia a ellos.
- Administración del almacenamiento secundario: Asigna espacio a los archivos en dispositivos de almacenamiento secundario.
- Mecanismos de integridad de los archivos. Garantiza que no se corrompa la información en un archivo. Solo debe estar la información que pertenezca al archivo.

En este tema abordaremos estos componentes.

Sistema de Archivos.

Métodos de acceso.

- *Acceso secuencial.* La lectura y escritura de registros se hace de forma consecutiva. Para hacer búsquedas se hace el recorrido desde el inicio del archivo hasta encontrar el registro requerido. La escritura de registros se hace al final.
- *Acceso aleatorio.* Se hace a través del valor de un campo del registro, el cual generalmente tiene un valor numérico entero que coincide con la posición dentro del archivo. Cuando el valor es muy grande, entonces se utilizan técnicas de dispersión (hashing). Por ejemplo, si se tiene un archivo de 500 registros de 75 bytes cada uno, y el usuario quiere acceder al registro 104, entonces el sistema de archivos puede colocar el apuntador de lectura del archivo al byte $75 * 104 = 7,800$. A este tipo de acceso también se le llama directo.
- *Acceso relativo a índice.* Permite almacenar registros de tamaño variable. La implementación consiste en dividir al conjunto de datos en dos secciones (posiblemente dos archivos). La primera sección es una lista corta de identificadores (números o nombres), cada uno con la ubicación de inicio y término del registro donde se encuentra el identificador que hace referencia de forma única al registro.

Sistema de Archivos.

Métodos de acceso (continuación).

- *Secuencial indizado.* Los registros se acomodan en secuencia lógica de acuerdo con una clave contenida en cada registro. El sistema mantiene un índice con las direcciones físicas de ciertos registros principales. El acceso a los registros secuenciales indizados se puede obtener secuencialmente por orden de clave o, de manera directa, mediante una búsqueda dentro del índice creado por el sistema.
- *De partición.* Es en esencia un archivo de subarchivos secuenciales. Cada archivo secuencial se llama miembro. La dirección inicial de cada miembro, se almacena en el directorio del archivo. Generalmente se utilizan los archivos de partición para almacenar grandes acervos.

Sistema de Archivos.

Métodos de consecución. Son los métodos que realizan la adquisición de bloques que se traerán a memoria principal o se grabarán en disco.

- *Por colas.* Se utiliza cuando es posible anticipar la secuencia en la que se procesarán los registros, como en el caso secuencial. Lleva a cabo una transferencia anticipada a buffers y una programación de las operaciones de E/S. De esta forma, procura tener el siguiente registro disponible para ser procesado tan pronto se procesa el registro anterior. Se mantiene más de un registro en almacenamiento primario a la vez, lo cual permite que se traslapen el procesamiento y operaciones de E/S (spooling).

También se encarga automáticamente de la agrupación de registros en bloque y la división del archivo en bloques.

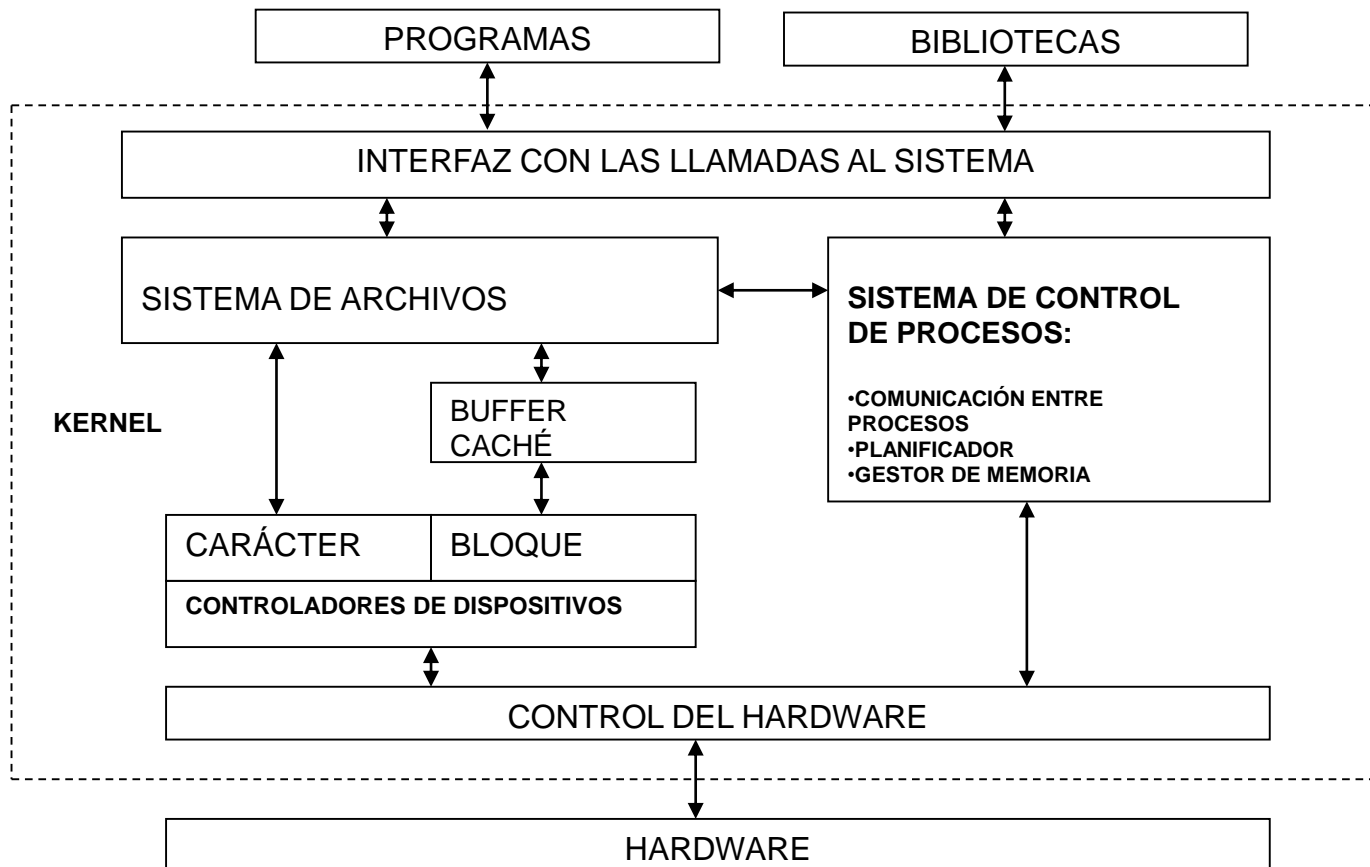
- *Básicos.* Se utiliza por lo regular cuando no es posible anticipar el orden en el que se van a procesar los registros, sobre todo en el caso de acceso aleatorio o directo. En este método, el usuario se encarga de la formación y división de bloques para los accesos de lectura y escritura de bloques físicos.

5 Sistema de Archivos

5.2 Caso de Estudio: Linux

El Sistema de Archivos en Linux (Parte 1).

Ahora revisemos algunas funciones del sistema de archivos de Linux, relacionadas a lo que hemos abordado hasta el momento. Primeramente ubiquemos al sistema de archivos en el kernel de Linux:



El Sistema de Archivos en Linux (Parte 1).

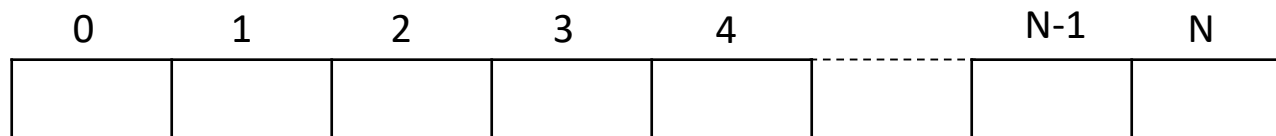
Podemos observar que el sistema de archivos se relaciona con varios elementos del Kernel.

- El elemento más en contacto con los programas del usuario es la interfaz de llamadas al sistema. Las cuales permiten realizar diversas operaciones sobre los archivos. Éstas son operaciones de *Alto Nivel*.
- Otro elemento de importancia con el que se relaciona, es el sistema de control de procesos, específicamente con el Gestor de memoria; con el cual realiza operaciones de transferencia de datos entre memoria principal y secundaria. Éstas son operaciones de *Nivel Intermedio*.
- Y también podemos observar que hay varios elementos del núcleo con los que se relaciona para finalmente acceder al dispositivo físico de almacenamiento secundario donde se encuentran los archivos. Dichos elementos están organizados en orden jerárquico, de los elementos más abstractos (como el buffer, el bloque, el carácter), hasta los elementos cercanos al hardware (como los controladores de dispositivos y el control de hardware). Las operaciones realizadas con estos elementos son de *Bajo Nivel*, ya que el objetivo final es el leer o escribir desde o hacia el disco.

El Sistema de Archivos en Linux (Parte 1).

Operaciones de bajo nivel.

- Linux considera a cada uno de los dispositivos periféricos como dispositivos lógicos, asociándoles un nombre de archivo. Es por esto, que para algunos dispositivos, como el monitor, hace la transferencia de datos, carácter por carácter. En nuestro caso nos referiremos al disco que, como vimos, la transferencia se hace por bloques.
- En el nivel más cercano al hardware están los controladores de dispositivos o *drives*, los cuales se encargan de ocultar al sistema de archivos las características hardware del disco. Es así que el sistema de archivos, percibe al disco como una secuencia numerada de bloques de tamaño fijo:



- Como ya sabemos, la lectura y escritura de datos que realiza el sistema de archivos al disco, se efectúa siempre en bloques de tamaño fijo. El driver proporciona primitivas para leer o escribir uno o varios bloques en cada operación.

El Sistema de Archivos en Linux (Parte 1).

Operaciones de bajo nivel.

- Sin embargo, trabajar con bloques de tamaño fijo, tiene como inconveniente de que exista fragmentación, si éste es muy grande o muchos accesos a disco si es muy pequeño. Es por esto, que el sistema de archivos de Linux maneja bloques lógicos que son de tamaño múltiplo del bloque físico y transferibles físicamente con una única llamada al manejador.
- Entonces, la asignación de bloques de datos a los archivos se realiza siempre en forma de bloques lógicos y si el último bloque no se llena con los datos del archivo, entonces se pueden asignar fragmentos de bloque lógico (fragmentSize) que a su vez son múltiplos del bloque físico.
- La relación entre tamaños de dichos bloques es la siguiente:
$$\text{Bloque físico} \leq \text{Fragmento} \leq \text{Bloque lógico}$$
- Los accesos a disco se realizan, siempre que sea posible, transfiriendo bloques lógicos. Sólo los últimos datos de cada archivo se asignan en forma de fragmentos para minimizar la fragmentación.
- Si el archivo crece, los fragmentos se pueden agrupar para formar bloques lógicos.

El Sistema de Archivos en Linux (Parte 1).

Operaciones de bajo nivel.

- Entre el sistema de archivos y el manejador de disco existe un elemento denominado *buffer caché*. La misión de este buffer es acelerar los accesos al disco tanto en modo lectura como en modo escritura. En este buffer caché se almacenan los últimos bloques que han sido utilizados.

Ejercicio 2

De acuerdo a la asignación de bloques de datos a los archivos, calcula cuántos accesos a disco deberá hacer el sistema de archivos de Linux para leer todos los datos para cada uno de los siguientes archivos. Calcula además el tamaño de fragmento interno en el último bloque transferido.

Datos:

Tamaño de bloque físico: 256 bytes

Tamaño de bloque lógico: 1,280 bytes

Tamaño de fragmento de bloque lógico: 512 bytes

Archivo	Tamaño archivo(bytes)
File1	1,730
File2	3,515

El Sistema de Archivos en Linux (Parte 1).

Ejercicio 2

Respuesta:

Hagamos primeramente el cálculo de cuántos bloques físicos entran en un bloque lógico y en un fragmento de bloque lógico.

Bloque lógico = $1280/256 = 5$ bloques físicos.

Fragmento de bloque lógico = $512/256 = 2$ bloques físicos

Para File1:

Cálculo de cuántos bloques físicos se le asignan: $\text{round}(1730/256) = 7$ bloques físicos.

Para leer todo el archivo se debe hacer un acceso de bloque lógico y uno de fragmento de bloque lógico. En este el último bloque (fragmento de bloque lógico) queda un fragmento interno de 62 bytes

Para File2:

Tiene asignados $3515/256 = 14$ bloques físicos.

Para leer todo el archivo se deben hacer 3 accesos de bloques lógicos, dejando un fragmento interno de 325 bytes en el último bloque lógico. Obsérvese que no se utiliza el fragmento de bloque lógico, porque no caben en él los últimos 4 bloques físicos. Considera que debe ser sólo un fragmento de bloque lógico. Si no, haría más transferencias.

Preguntas base para siguiente examen. Contestarlas en sus notas.

- Identificar las funciones del sistema de archivos en aseveraciones.
- Distinguir las características de las diferentes organizaciones de archivos.
- Ventajas y desventajas de los diferentes métodos de acceso. Es decir, a la forma en que se obtiene acceso a los datos almacenados en archivos.

Ejercicio A

Un cierto sistema de archivos maneja un registro físico o bloque de 2048 bytes; si tiene que hacer la lectura completa de los siguientes archivos, indica por cada archivo cuántas transferencias tiene que hacer, si se tratan de registro sin bloques o en bloques y si se presentan fragmentos.

Archivo	Tamaño archivo(bytes)	Tamaño registro lógico
Arch1	4,200	120
Arch2	9,450	45

Preguntas base para siguiente examen (continuación). Contestarlas en sus notas.

Del Sistema de archivos de Linux:

Identificar las funciones del sistema de archivos en aseveraciones.

- Distinguir, de una lista, las operaciones que hace el sistema de archivos en cada nivel.
- ¿Qué método de consecución realiza el sistema de archivos de Linux? ¿Por qué?

Ejercicio B

De acuerdo a la asignación de bloques de datos a los archivos, calcula cuántos accesos a disco deberá hacer el sistema de archivos de Linux para leer todos los datos para cada uno de los siguientes archivos. Calcula además el tamaño de fragmento interno en el último bloque transferido.

Datos:

Tamaño de bloque físico: 128 bytes

Tamaño de bloque lógico: 1,024 bytes

Tamaño de fragmento de bloque lógico: 640 bytes

Archivo	Tamaño archivo(bytes)
File1	4,430
File2	2,715